Advances in Computer Science and Engineering

Research in Computing Science

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 19**, Mayo, 2006. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. 04-2004-062613250000-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor Responsable: *Juan Humberto Sossa Azuela,* RFC *SOAJ560723*

***Research in Computing Science** is published by the Center for Computing Research of IPN. **Volume 19**, Mayo, 2006. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, Mayo, 2006, in the IPN Graphic Workshop – Publication Office.*

Volume 19

Volumen 19

Advances in
Computer Science and Engineering

Volume Editors:

Editores del Volumen

Alexander Gelbukh
Sulema Torres
Itzamá López

Instituto Politécnico Nacional
Centro de Investigación en Computación
México 2006

ISSN: 1665-9899

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.ipn.mx
http://www.cic.ipn.mx

Printing: 500
Impresiones: 500

Printed in Mexico
Impreso en México

Preface

Information technologies are one of the most influencing, dynamic, and rapidly growing areas of human activity nowadays. The information technology research can be subdivided into two major areas: computer science and computer engineering. Computer science studies the algorithms and data structures used for logical organization of computer programs, while computer engineering is aimed on development of better and faster devices that serve as the physical basis for execution of the programs. Finally, both areas collaborate on development of practical applications that improve the quality of people's life.

This volume contains 20 carefully selected internationally peer-reviewed and revised original research papers on both theoretical advances and practical applications of computer science and engineering. The papers are structured into the following six sections:

- Software Technology
- Information Systems
- Networking
- Logic and String Algorithms
- Image Processing
- Applications

The volume will be useful for researches, engineers, and students working in the respective areas of computer science and engineering, as well as for all readers interested in computer science, computer engineering, and real-life applications of computers.

This volume is a result of work of many people. In the first place we thank the authors of the papers included in this volume, for it is the technical excellence of their papers that gives it value. We thank also the members of the International Editorial Board of the volume and the additional reviewers for their hard work on selecting the best papers out of many submissions we received. We would like to thank Edgar Gatalán Salgado and Alejandro Cuevas Urbina, as well as the personnel of the Center for Computing Research of the National Polytechnic Institute, in the first place Oralia del Carmen Pérez Orozco and Ignacio García Araoz, for their indispensable help in the preparation of the volume, with special thanks to Israel Román.

With this volume and the next one we commemorate the 10th anniversary of the Center for Computing Research of the National Polytechnic Institute, the 70th anniversary of the National Polytechnic Institute, and the 15th anniversary of CIC, the International Conference on Computing.

May, 2006

Alexander Gelbukh
Sulema Torres
Itzamá López

Table of Contents
Índice

Page/Pág.

Software Technology

Information Systems

Networking

Logic and String Algorithms

Image Processing

Applications

Software Technology

Pre-conceptual Schema: a UML Isomorphism for Automatically Obtaining UML Conceptual Schemas

Carlos Mario Zapata Jaramillo,[1] Fernando Arango Isaza,[1] Alexander Gelbukh [2]

Universidad Nacional de Colombia,
Carrera 80 No. 65-223 Oficina M8-112 Medellin, Colombia
{cmzapata, farango}@unalmed.edu.co

Computing Research Center (CIC), National Polytechnic Institute,
Col. Zacatenco, 07738, DF, Mexico
www.Gelbukh.com

Abstract. Software development methodologies improve model quality. Conceptual schemas are representations of the universe of discourse for development purposes. UML has become a de-facto standard in software modeling. Obtaining UML diagrams from natural language descriptions is a very attractive goal. In this paper, we present a proposal that improves some drawbacks from the previous work on this area. We call the proposed representation a pre-conceptual schema. It is an intermediate stage between natural language and UML conceptual schemas. Finally, we show a case study for rules applying.

1 Introduction

Quality has been increasingly important in software development; from the point of view of Software Engineering, quality in software is the result of the application of a disciplined and methodological approach, covering all aspects of software life cycle [1].

Through a methodological approach, models have capital importance, because they permit specification understanding, communication among members of development team, future maintenance of the system, and reuse of code and specifications. In this methodological approach, UML had become a de-facto standard for software development [2].

Two trends in Software Engineering have become to gain importance for analysts:

- Firstly, CASE tools have improved capabilities of analysts to create UML diagrams. The main goal of CASE tools is support for drawing and editing diagrams, but responsibility for domain knowledge and obtaining of UML diagrams from natural language is left to analysts [3].
- Secondly, there is an increasing wave for the automated obtaining of UML diagrams. In this trend, responsibility for domain knowledge and obtaining of UML diagrams from natural language is left to computers, and theoretical studies have been made for rules definition to achieve this goal.

© A. Gelbukh, S. Torres, I. López (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 19, 2006, pp. 3-14

There are several works on the second trend [4–9], but certain problems are still open. Generally speaking, many proposals in this trend tends to work on only a kind of diagram (Entity-Relationship Model, Class diagram, and so on), but in the presence of two or more diagrams, consistency problems are common.

In this paper, adopting the second trend, we propose an approach for automated UML diagrams obtaining through the use of a new graph named Pre-conceptual Schema and a set of translation rules.

This paper is organized as follows: in Section 2 we survey some works on automated UML obtaining; then, in Section 3 we describe Pre-conceptual Schema as a new approach to this issue. Section 4 is devoted to the definition of rules for automatic transformation from Pre-conceptual Schemas and three kinds of UML diagrams. A case study in Spanish language is developed in Section 5, and in Sections 6 and 7 we discuss some conclusions and future works, respectively.

2 Automated Obtaining of UML Diagrams: A Survey

Many researchers in the world are trying to obtain UML conceptual schemas in an automated way from natural language. This trend has been a dream for modelers since the early attempts of Peter Chen, the father of Entity – Relationship Model (ERM) [10], who defined a set of rules to obtain ERM from an English discourse [11]. They were simple rules, but it was possible to find many counter-examples for each of them; for this reason, Chen named them “suggestions” more than “rules”. In the same way of thinking, Coad and Yourdon defined another set of rules for Class Diagram, in the early years of object-orientation [12]. Neither Chen nor Coad and Yourdon had the intention to automate their rules, but they generated the basis for a new research in modeling.

Progress in Software Engineering has increased development and use of many new tools, named Computer-Aided Software Engineering (CASE) Tools [3]; for modelers, these tools have become electronic assistants for model drawing. However, CASE technology has been founded under the assumption that modelers have to interpret the domain of discourse and they may convert natural language specifications into the required diagrams, and then they can use CASE tools for drawing these diagrams. Automated assistance begins in this moment, but there’s no help in previous stages of the process.

A semi-automated approach has been developed by LInguistic assistant for Domain Analysis (LIDA) Project [4]. In LIDA, a classification for kinds of words is made along a discourse in natural language; LIDA identifies nouns, verbs, and adjectives, and it calculates frequencies of word’s appearance in the text. With this information at hand, modeler must decide if the word will be mapped to a class, an attribute, an operation or a relationship in the class diagram. Mapping process is, therefore, owned by the modeler, with little assistance of the LIDA tool.

Rapid Application and Database Development (RADD) Project [5] was designed to obtain ERM from natural language in an automated way, and initiating a “moderated” dialogue to enhance completeness of the diagram. However, RADD was designed for ERM, and its creators didn't define mapping for another kinds of conceptual schemas, e.g. UML diagrams.

A different approach was defined by Cyre in Automatic Specification Interpreter (ASPIN) Project [6]. In ASPIN, modeler can create multiple diagrams (e.g. timing, State-Transition, Blocks, etc.) for describing a control system specification, and then ASPIN can create a consolidated representation of the system, based on those diagrams. Although ASPIN doesn't use natural language (it only accepts a restricted form of language, specific to the domain of control systems), its approach is useful for demonstrating the possibility of joining diagrams together in a single representation (it uses Conceptual Graphs for this purpose). However, ASPIN only works with control systems domain, and lacks generality for working with another paradigm (such as UML, for example).

CM-BUILDER project [8] was developed for automated UML class diagram acquisition. The process begins from natural language specifications, but it requires a previous knowledge about domain of the problem. This knowledge must be represented through semantic nets, with almost every category of the class diagram in them. Semantic nets, like these, are very complex to acquire, and they don't guarantee mapping process if a word doesn't match a category in them.

NL-OOPS (Natural Language Object – Oriented Product System) Project [7] uses a semantic net for mapping process too. However, rules used by the process lack of generality, and identified elements must belong to several categories simultaneously. Therefore, mapping process requires active participation of the analyst, who must decide the final category for every element.

Both CM-BUILDER and NL-OOPS were developed for UML class diagram acquisition and they don't identify elements for another UML diagrams. On the opposite, NIBA Project [9] was developed for multiple UML diagrams acquisition (mainly class and activity diagrams). It uses the so called KCPM (Klagenfurt Conceptual Predesign Model), a model with various kinds of elements to achieve mapping process, from tables to certain dynamic diagrams. KCPM is not unified, and its use depends on the kind of target diagram; as a consequence, it's difficult to guarantee consistency between diagrams, because every element for every diagram comes from different forms of KCPM.

Work has been done in this area, but problems still remain; particularly, problems are unsolved for consistency reasons, standardization (UML is a standard, but many works try to obtain some other formalism), and connectivity between formalisms. In the next section, we define a new proposal for an intermediate stage between natural language and UML conceptual Schema: Pre-conceptual Schema.

3 Suggested Approach: Pre-conceptual Schema

3.1 Justification

Some of the works listed in the previous section uses an intermediate formalism, in order to facilitate mapping process. Semantic nets, tables, dynamic graphs and conceptual graphs are some of the mechanisms used by these projects for representing natural language discourse.

An intermediate formalism acts as a facilitator of the mapping process between natural languages and UML diagrams. It's needed because natural language representation lacks of certain elements and relationships required for mapping process.

3.2 Definition

In this paper we use a new kind of intermediate formalism, and we call it "Pre-conceptual Schema". The term "pre-conceptual" was coined by Heidegger [13] and it refers to previous information, acquired in some way, of a concept. In the knowledge stages, Piaget [14] identifies a stage, later to linguistic knowledge, but previous to conceptual knowledge, and he called it "pre-conceptual stage".

In some way, we are trying to gather some useful information for mapping process from natural language to UML diagrams. To achieve this goal, we need an intermediate stage, founded in the linguistic information, but with certain knowledge of the later phase of the process: the conceptual one. As a consequence, our approach needs a new schema, an intermediate schema for facilitating mapping process between natural language specifications and UML diagrams. We called it "Pre-conceptual Schema" PS because of the above reasons. Our main goal is the definition of PS syntax and to prove UML diagrams can be contained in PSs.

PS must accomplish three asserts:

- PS must be obtainable from natural language (demonstration is out-of-scope of this paper).
- PS must be isomorphic with UML diagrams (through the set of rules defined in section 4).
- PS must be rewritable in a disambiguated form of language, a simple discourse with little or no ambiguities.

Acting as an isomorphism, PS must represent generalities of UML diagrams, as an integrated view of the same model. For this reason, consistency problems must disappear, because starting point for every diagram drawing is the same.

3.3 Notation

In ASPIN [6], they used Conceptual Graphs CG as an intermediate formalism for representation of many diagrams simultaneously. The reasons for this election are representativeness and versatility. But, thinking about our goals, Conceptual Graphs poses some drawbacks for us:

- Representation of a concept, which is included in various phrases simultaneously, is complicated. For this goal, CGs use a dotted line called "correference"; however, this mechanism is more intricate as far as the same concept appears over and over again.

- CGs are preeminently structural graphs. For the sake of variety, we need a schema capable of representing both structural and dynamic properties, and CGs only can represent structural features.
- Usage of a symbol for many kinds of representations produces ambiguity. In CGs concepts are used for both nouns, verbs and adjectives, and relationships are used for both semantic cases and other semantic relations.

Pre-conceptual Schemas are founded on CGs, but we have made some changes for supporting main features of the mapping process. The main symbols in Pre-conceptual Schemas are Concepts (rectangles) and relationships (ovals), but differing from CGs, in concepts we can only put nouns, and in relationships we can only put verbs. If a concept is repeated too many times in a discourse, in PSs will appear only once; furthermore, all the relationships to this word in the discourse will be represented as relationships with this only one concept. In this way, PSs will be integrated graphs and they will not be like many CGs with co-referents.

For dynamic purposes, in PSs there are two new elements:

- Thick arrows express implication. These elements only can be connected from one relationship to another, and it means the target verb it's only performed if the source verb is performed (as a precondition in if-then phrases).
- Rhombs express conditionals. In the inner space of the rhomb, we can put expressions with concepts and operators; these expressions must be true or false.

Similarly to CGs, in PSs thin arrows represents directed connections. In PSs, They must be:

- Concept-relationship connection: it means one concept achieves an activity expressed by the relationship.
- Relationship-concept connection: it means one concept receives an activity expressed by the relationship. In this case, we can include a preposition in the arrow (if it's needed)
- Conditional-relationship connection: it means one activity (expressed by the relationship) will be executed if the answer for the conditional matches the word included in the arrow (we must include that word).

In directed connections, we can include a number for distinguishing actions before and after an implication occurs.

In Figure 1 we can see the main symbols of PSs.

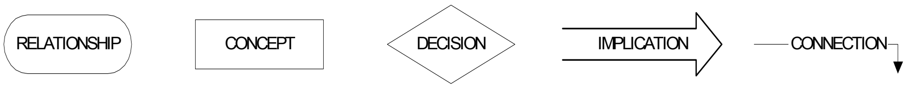


Fig. 1. Main symbols of Pre-conceptual Schema

3.4 Additional Considerations

In order to define and use properly PSs, we must warn you about two special considerations:

- We have only defined the main symbols used in PSs. Users of PSs must decide the best way to express their needs in terms of them.
- Concepts admit the use of compound nouns. Again, user must decide usage.

In the next section, we define a set of rules for automated transformation between PSs and three kinds of UML 2.0 diagrams: class, communication (in previous versions of UML, it was called “collaboration” diagram) and state machine diagrams.

4 Rules for Automated Transformation from Pre-conceptual Schema to UML 2.0 Diagrams

In the previous section, we defined main features of PSs as an intermediate formalism to perform transformation between natural language specification and UML diagrams. Rules for obtaining PSs from natural language are out-of-scope of this paper. Instead of, we must present rules for transformation from PSs to three UML diagrams, for Spanish language (some rules can be different in English language).

4.1 Rules for Class Diagram

1. A source concept from a “has” relationship is a candidate class.
2. A target concept from a “has” relationship is a candidate attribute.
3. Both the source and target concepts from an “is” relationship are candidate classes (an exception is made if one or both concepts are adjectives or proper nouns). The relationship itself is a candidate inheritance with source concept as a candidate daughter class, and target concept as candidate parent class.
4. A concept defined as a candidate class by one or more rules, and as a candidate attribute by another set of rules, is a class.
5. Relationships corresponding to activity verbs or realization verbs are candidate operations of target concepts (if these concepts are defined as candidate attributes by one or more rules, the operations are assigned to their owner candidate classes).
6. A candidate operation between two classes generates a candidate association between these classes.
7. A “has” relationship between two concepts, which are identified as candidate classes by one or more rules, generates a candidate aggregation relationship, with source concept as the whole and target concept as the part.
8. Concepts identified as object classes in communication diagram are candidate classes in class diagram.
9. Relationships identified as messages between objects in communication diagram are candidate operations of target object class in class diagram.

4.2 Rules for Communication Diagram

1. The source set of concepts and relationships from an implication connection is a candidate guard conditions.
2. Expressions included in conditionals are candidate guard conditions.
3. Target relationships after either an implication or a conditional are messages. The source concept will be source object class and target concept will be target object class. There must be series of messages jointed by objects.

4.3 Rules for State Machine Diagram

1. Past participle messages identified in communication diagram are candidate states for target object class.
2. Sequence between states in State Machine Diagrams depends on identified and numbered sequences in communication diagrams.

5 A Case Study: Application of the Rule to Spanish Language

We developed the rules described in Section 4 for Spanish language, because our research group is trying to obtain conceptual diagrams from specifications in this language. The case study relates to a pizzeria and its production and delivery processes. In Figure 2 we can see Pre-conceptual Schema from this domain, and in Table 1 we summarize the application of some rules. The application of the rules for this case study was hand-made for academic purposes; as a future work, we are planning to build a CASE tool with the automation of this method.

Some pieces of this diagram must be rewritten as follows:

- Clients are persons.
- Details have a quantity, an observation and a product.
- If the difference between delivery hour and exit hour is lower than 30 minutes, then dispatcher registers payment, else dispatcher makes a devolution report.
- Whenever client calls, dispatcher registers order.

Note how this way of textual representation of PSs must express, in a cumulative way, the Universe of Discourse associated with a model of the world.

Table 1 shows how product is initially defined as a candidate attribute by rule 2 from Section 4.1 and then redefined as a candidate class by rule 4. Furthermore, in UML associated element appears some elements needed for the element definition; e.g. “Client” needs “Person” for its inheritance, and “deliver” needs source object class (“deliverer”) and target object class (“order”).

With the hand-made application of rules described in section 4, we must obtain diagrams showed in Figures 3, 4, and 5.

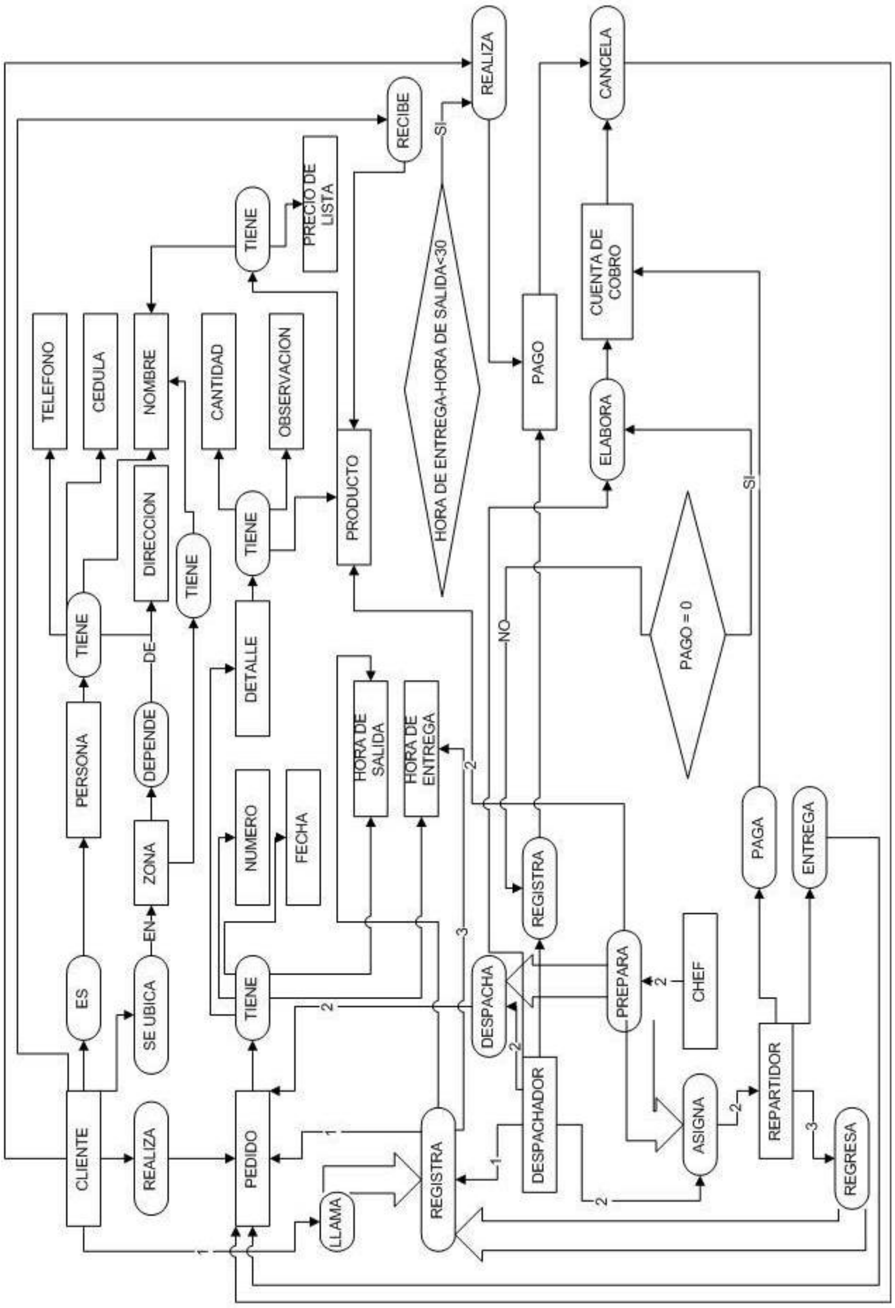


Fig. 2. Pre-conceptual Schema of production and delivery process of a pizzeria.

Table 1. Rules application for some elements of PS.

PS Element	UML Diagram	UML Element	UML associated element	Rule (Section)
Client	Class	Candidate Class	Person	1, 3 (4.1)
Order	Class	Candidate Class		1 (4.1)
Product	Class	Candidate Class		2, 4 (4.1)
Address	Class	Candidate Attribute	Person	2 (4.1)
Register	Class	Candidate Operation	Order	5, 9 (4.1)
Chef prepares product	Communication	Guard Condition	Assign	1 (4.2)
Deliver	Communication	Candidate Message	Deliverer, Order	3 (4.2)
Delivered	State Machine	Candidate State	Order	1 (4.3)

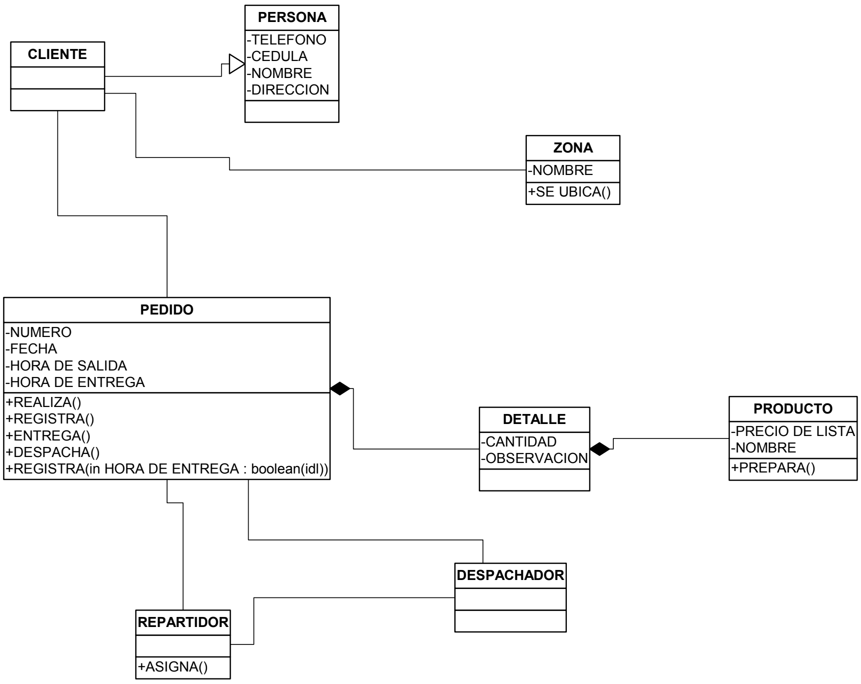


Fig. 3. Resultant Class Diagram from PS in the Figure 2.

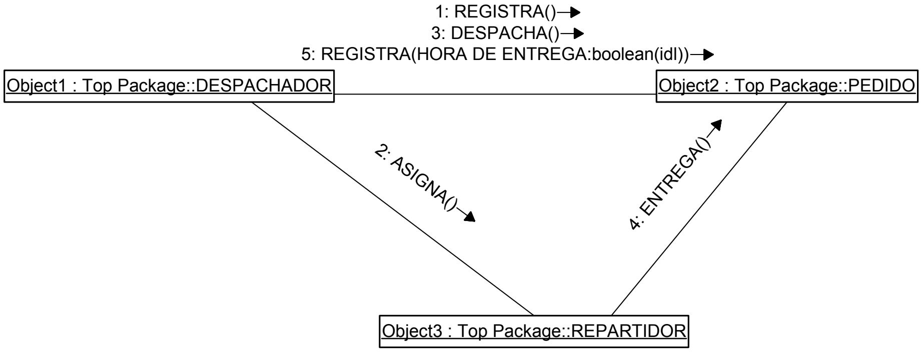

1: REGISTRA()
3: DESPACHA()
5: REGISTRA(HORA DE ENTREGA:boolean(idl))
Object1 : Top Package::DESPACHADOR
Object2 : Top Package::PEDIDO
2: ASIGNA()
4: ENTREGA()
Object3 : Top Package::REPARTIDOR

Fig. 4. Communication Diagram from PS in the Figure 2.

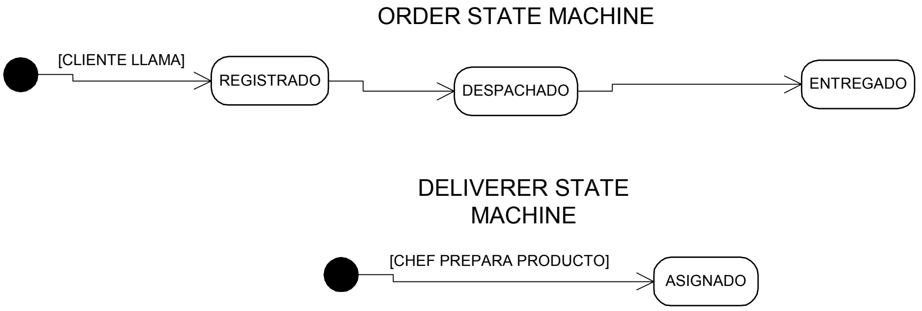


Fig. 5. Resultant State Machine Diagrams from PS in the Figure 2.

6 Conclusions

In this paper we have presented Pre-conceptual Schemes, an intermediate stage between natural language specifications and UML diagrams. Furthermore, we have developed a set of rules for automated obtaining of three kinds of UML diagrams: class, state machine, and communication diagrams. We have assumed Pre-conceptual Scheme as being obtainable from natural language (demonstration is out-of-scope of this paper), and we have concentrated in the next transformation stage.

With the case study, we showed it's possible to obtain those UML diagrams (class, state machine, and communication diagrams) with some limitations, but with the main features of the diagrams. Pre-conceptual Schemas are even very simple, and they represent restricted natural language specifications; however, they lack of mechanisms for representing words like adverbs and adjectives. Even with limitations, they can describe a domain of discourse in a complete and understandable way.

7 Future Work

- As a continuation of this work, it will be needed the definition of additional sets of rules for automated obtaining of UML diagrams (e.g. timing or activity diagrams), and additional rules for the diagrams here described (e.g. rules for multiplicity of associations in Class diagram, rules for "on exit" actions in state machine diagram, and so on). Furthermore, we must develop a prototype to prove application of this work in the second trend described by section 1 (introduction).
- We must generate rules for Pre-conceptual Scheme obtaining from natural language.
- Furthermore, we must generate new mechanisms for representation in Pre-conceptual Schemas (e.g. adverbs and adjectives) for extending their functionality and scope.

References

1. Pressman, R.: Software Engineering: A Practitioners' Approach. 5th edn, McGraw-Hill, Inc, New York (2001)
2. OMG: UML Specification. Available: http://www.omg.org/uml
3. Burkhard, D. and Jenster, P.: Applications of Computer-Aided Software Engineering Tools: Survey of Current and Prospective Users. Data Base Vol. 20, No. 3 (1989) 28-37
4. Overmyer, S.P., Lavoie, B., y Rambow, O.: Conceptual modeling through linguistic analysis using LIDA. In: Proceedings of ICSE 2001, Toronto, Canada. (2001)
5. Buchholz, E. y Düsterhöft, A.: Using Natural Language for Database Design. In: Proceedings Deutsche Jahrestagung für Künstliche Intelligenz. (1994)
6. Cyre, W.: A requirements sublanguage for automated analysis. International Journal of Intelligent Systems, Vol. 10, No. 7. (1995) 665-689.
7. Mich L.: NL-OOPS: From Natural Natural Language to Object Oriented Requirements using the Natural Language Processing System LOLITA. Journal of Natural Language Engineering, Cambridge University Press, Vol. 2, No. 2. (1996) 161-187.
8. Harmain, H. y Gaizauskas, R. : CM-Builder: An Automated NL-based CASE Tool. In: Proceedings of the fifteenth IEEE International Conference on Automated Software Engineering (ASE'00), Grenoble. (2000)
9. NIBA Project.: Linguistically Based Requirements Engineering - The NIBA Project. In: Proceedings 4th Int. Conference NLDB'99 Applications of Natural Language to Information Systems, Klagenfurt. (1999) 177 – 182.
10. Chen, P. P.: The Entity–Relationship Model: Toward a Unified View of Data. ACM Transactions on DataBase Systems, Vol. 1, No. 1. (1976)
11. Chen, P. P.: English Sentence Structure and Entity–Relationship Diagrams. Information Science, No. 29, Vol. 2. (1983) 127-149.
12. Coad, P. y Yourdon, E.: Object – Oriented Analysis. New Jersey: Yourdon Press. (1990)
13. Heidegger. M.: Protokoll zu einem Seminar über den Vortrag "Zeit und Sein". En: Zur Sache des Denkens, Tübingen (1976) 34.
14. Piaget, J.: The origins of intelligence in children (2nd ed.). New York: International Universities Press (1952)

Web Services Procurement Based on WSLAs

Giner Alor-Hernandez [1], Juan Miguel Gomez [2]

[1]Division of Research and Postgraduate Studies
Instituto Tecnologico de Orizaba.
Av. Instituto Tecnologico 852, Col Emiliano Zapata. 09340 Orizaba, Veracruz, México.
e-mail: galor@itorizaba.edu.mx

[2]Departamento de Informática
Escuela Politécnica Superior, Universidad Calos III de Madrid.
e-mail: juanmiguel.gomez@uc3m.es

Abstract. This paper describes a framework for providing differentiated levels of Web services to different customers on the basis of service level agreements (SLAs). Under the framework described in this paper, service providers can offer Web services at different service levels. In general, the service levels are differentiated based on many variables such as responsiveness, availability, and performance. The framework comprises the Web Service Level Agreement (WSLA) language to specify SLAs in a flexible and individualized way, a system to monitor the compliance of a provided service with a service level agreement, and a workload management system that prioritizes requests according to the associated SLAs.

1 Introduction

A Web service is a software component that is accessible by means of messages sent using standard web protocols, notations and naming conventions, including the XML protocol [1]. The notorious success that the application of the Web service technology has achieved in B2B e-Commerce has also lead to consider it as a promising technology for designing and building effective business collaboration in supply chains. Deploying Web services reduces the integration costs and brings in the required infrastructure for business automation, obtaining a quality of service that could not be achieved otherwise [2], [3]. Therefore, Web services offer a new way for the development of distributed applications which can integrate any group of services on the Internet into a single solution. It may involve, possibly, the use of web services provided by different organizations, cooperating in complex collaborations. Thus, there is a need of agreements in order to establish the obligations to both sides, i.e. customers which use Web services and providers which supply them. Commonly, these agreements are defined by using Web Service Level Agreement (WSLA) Language. A WSLA document defines assertions of a service provider to perform a service

© A. Gelbukh, S. Torres, I. López (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 19, 2006, pp. 15-28

according to agreed guarantees for IT-level and business process-level service parameters such as response time and throughput, and measures to be taken in case of deviation and failure to meet the asserted service guarantees [4]. The assertions of the service provider are based on a detailed definition of the service parameters including how basic metrics are to be measured in systems and how they are aggregated into composite metrics [5]. In addition, a WSLA expresses which party monitors the service, third parties that contribute to the measurement of metrics, supervision of guarantees or even the management of deviations of service guarantees [6]. Interactions among the parties supervising the WSLA are also defined. Having this into account, we have developed a framework for Web Services procurement which provides needed functionalities to business services and allows developers of business services to focus on their business domains rather than on support issues. Our framework features provisioning services, such as contracting, metering, accounting, notification and SLA based management of web services.

The rest of this paper is structured as follows. In the next section we present the general architecture and its main components of the framework for Web services procurement. In the following sections, we discuss the functionality of each component of the service Hub which is the main component in our architecture and discuss their relationships among them. Next, we present a scenario for Web services procurement among services requestors and providers. Then we describe the future directions and review the related work. Finally, we emphasize the contributions of our work.

2 Architecture

Our architecture was designed to show how to extend a Web Service by adding common provisioning services that most business infrastructure will need. Under our architecture, there are four steps that occur during the processing to carry out the procurement of Web services:

1. A Business Service (Web Service) is hosted on a machine. This service is registered with a Service Hub.
2. A Hub administrator will create an Offering package. An Offering is a contract that specifies what service is being made available and at what performance levels.
3. A Service Requestor will then choose an Offering package, thus creating a Usage Contract - agreeing to the terms of the contract - including the billing terms.
4. Once the Usage Contract is activated the requestor is now free to use the Service.

Fig. 1 shows a multi tier configuration of our framework for procurement of Web services. Below is a high-level view of the configuration. In Fig.1, the first tier is the Service Requestor (Client). In this tier, an initial SOAP request is generated which is sent to the second tier - a Service Provider. This message passes through an Axis handler which places the identity of the Requestor into the SOAP message so that the

Service Provider can properly identify who is trying to use the service. In the second tier, the service Hub, acts as a gateway to the services being offered. A hub administrator is responsible for registering and managing the services that are available from this service hub. The service hub has the responsibility of invoking all of the provisioning services and then ultimately routes the request to the desired Supplier to actually do the business logic of the Web Service. The internals of the service hub are shown in Fig. 2.

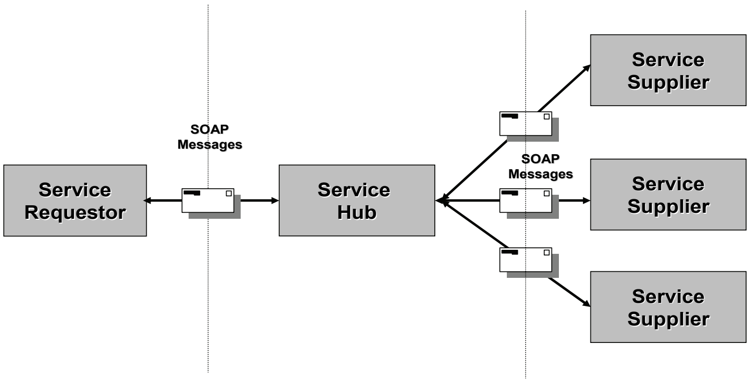


Fig. 1 A multi tier configuration of the framework for Web services procurement

When a SOAP message enters the service hub (as shown in Fig. 2), the Axis servlet passes it through a series of handlers responsible for interacting with each of the provisioning services. First a profile handler uses the Profile Service to validate the Service Requestor identity and gets its unique profile key. This key is stored in the SOAP message context so that it can be accessed by other handlers. Then, the contract handler invokes the Contract Service to verify that the service requestor has a valid Usage Contract and places the contract ID into the message context. Next, the metering request handler generates a start metering event, which is used for accounting purposes. The management request handler loges the request for statistical purposes. Next, the request is processed by the Web Services Management Middleware (WSMM) handler. This handler takes the contract ID from the message context and retrieves the WSLA performance expectation data from the contract. Based on this data, plus the load on the machines hosting the business services, the WSMM handler determines when to allow the request to continue down the chain of handlers. Finally, the message is passed to the Service Desk handler which routes the request to the proper machine hosting the Business Service. The WSMM handler acting in conjunction with the Service Desk acts as a load balancing mechanism - to help ensure all of the performance criteria expected by the contracts are met.

On the output side, the response from the Business Service is processed by the Metering, Management and WSMM Response handlers. Each one uses its specific service to make a note of the completion of the Business Service's processing. The response message is returned to the Service Requestor.

The third tier, the Service Supplier, is a machine hosting the actual Web Service. This machine does not need any special set-up beyond the normal Web Services configuration (SOAP server and the Business Service itself). The Hub machine must be aware of its existence in order for this supplier can participate. A hub administrator is responsible for handling this as part of the process of managing the services available from the service hub.

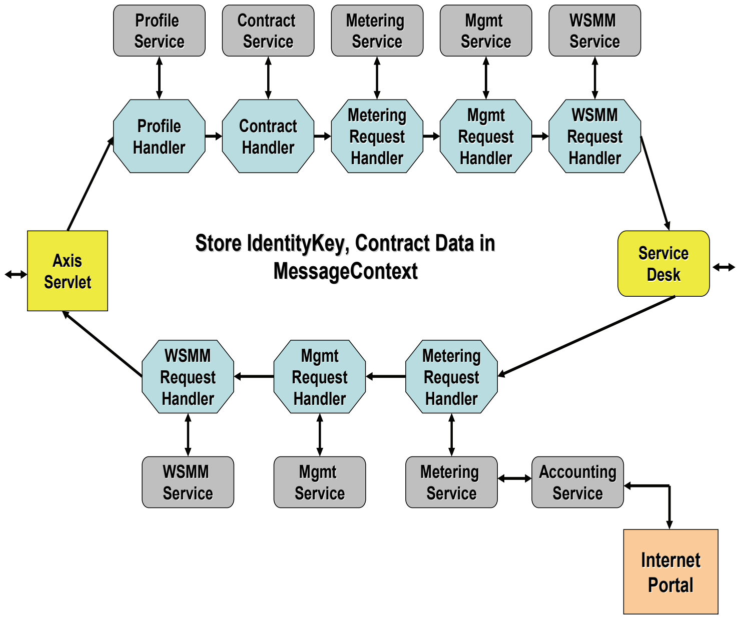


Fig. 2 Internals of the service hub

Managing the services available from the service hub involves registering the interface for a service that is available and then registering any Suppliers that actually provide the implementation for this service interface. Once a service is registered, it can be made available to requesters through either fixed of flexible offering packages. A service can be used as long as there are any Service Suppliers registered that provide the service. The service hub manages which Supplier handles requests for a service.

Based on this understanding, our approach is based on offerings and usage contracts. An offering is created by the Hub administrator and indicates which business services are available to a service requestor and what performance guarantees it is offering for that service. The service requestor establishes a usage contract before using a business service. Furthermore, the user profiles for the service requestor and hub administrator must be registered.

In the next section, we describe with more detail the functionality of each internal from the service Hub and their relationships among them.

3 Internals of the Service Hub

3.1 Compliance Monitor

The Compliance Monitor supervises whether the service level objectives specified in a WSLA document are met and raises an alarm otherwise. Upon activation of a new usage contract and its corresponding WSLA, the relevant performance data is read from the metering service, aggregated as defined in the WSLA document and the service level objectives are evaluated. If violations or other relevant conditions occur, as defined in the action guarantees, actions such as notifications can be taken.

The compliance monitor consists of three functional components:

1. A data provider connects to the source of measurement, e.g., the instrumentation or in our case the measurement service, and reads raw measurements according to the measurement directives of a WSLA document.
2. The data provider is invoked by a measurement component. This component interprets the SLA parameter and metric statements of a WSLA document and aggregates the raw metrics as retrieved by the data provider to SLA parameters according to the WSLA specification.
3. New SLA parameter values are forwarded to the compliance monitor component. This component evaluates the conditions of the service level objectives and executes the activities defined in the action guarantees of the WSLA document, typically sending notifications to the notification service.

The compliance monitor provides the following operations:

- **add** - add a WSLA to be monitored
- **remove** - stop monitoring a WSLA and remove it
- **getActive** - get the list of currently monitored WSLAs

3.2 Web Services Management Middleware (WSMM)

WSMM is a feedback control mechanism for transparent performance management of web services, in a way that maximizes expected business value in the face of service level agreements (SLAs) and fluctuating offered load. The main objective of WSMM is to provide support for differentiated services based on Service Level Agreements (SLAs). The introduced mechanisms enable service providers to offer the same web service at different performance levels (e.g., different average response time thresholds), in a way that is transparent to the service and client developers.

WSMM performs resource allocation, scheduling, and overload protection by means of a collection of real-time mechanisms, which are applied to individual requests, as well as slower time scale optimization and coordination mechanisms.

3.3 Service Desk

Web Services binding, interoperability, sharing, and aggregation of multiple heterogeneous Web Services are key integration problems. We have used Service Desk technology [7] that provides intelligent Web Services clustering. Service desk is a specific service domain object described in WSDL documents that represents the basic processing unit of a collection of services. Service Desk technology allows create, cluster, organize, route, recover, and switch Web Services in an autonomous way [7]. The cluster can represent a group of comparable or related services through a common services entry point, in fact a service grid. Subsequently, responsive to the receipt of service requests, the grid can select suitable ones of the computing services instances to process the received service requests, monitor the performance of the selected instances, and perform fail-over processing if required. The selection is according not only to availability, but also according to QoS characteristics, as specified via WSLAs, and business arrangements. The operation is automatically performed based on a service policy. The set up process follows the eUtility model [8] of creation of service offering, and customer subscription, for both the service desk clients and service desk suppliers. This technology demonstrates an integration benefit of Web Services, Autonomic computing and Grid computing utilized as a whole.

3.4 Metering Service

The Metering Service receives meter events from clients and provides meter events upon request. The Accounting Service gets metering information from the Metering Service and contract information from the Contract Service and uses this to produce a usage report for a particular client using a particular service. The Metering Service supports three types of WSDL-defined operations from a client: (1) recordMeterEvent, (2) recordMeterEvents, and (3) getMeterEvents.

Metering is possible on an operation level. Meter events contain the service name and the operation name of the service that was called, timestamps, as well as the id of the contract used to handle the request. Meter events vary by type so various ways of charging a service call are possible (specified in the service contract set up by the Contract Service):

- Start/end events are used when access to a service is charged by the amount of time used to perform the service;
- Ad-hoc events are used when access is charged for by the number of times that the service is accessed, or on some other basis besides time.

In addition to the above, two more types of events are available: cancelled, which is used to cancel an event which has already been sent to the metering service, and unknown, which is used when the type of event was not supplied by the service requestor.

3.5. Contract Service

The Contract Service handles the relationship between service providers and service requestors. It provides information about the type of contract between a service provider and the service hub (deployment contracts) and between a service requestor and the service hub (usage contracts). Usage contracts can be used to subscribe to any combination of operations of any service provided through the service hub. They can include fixed services, or allow for services to be added or removed over the life of the contract. A usage contract contains information such as how calls to service operations are to be charged for (by time, by number of uses, among others) and how much the subscribed service operations should cost for that client. For each usage contract the Contract Service defines the payment model and rating model to be used, the effective dates for that contract. Contracts may optionally store the digital signatures of both parties (service hub and service provider/requestor) to the contract. Under our approach, contracts are added to the Contract Service via our framework and a valid contract must be in place between a service hub and a service requestor before the requestor can use the service. The Contract Service supports WSDL-defined operations such as the following: (1) createContract, (2) getContractModel, (3) getContractState, (4) updateConstractState, (5) getContractType, (6) setContract-Property, (7) getContractProperty, and (8) getUsgaeContractsValidForIdentity.

3.6 Accounting Service

The Accounting Service is used to calculate billing data according to rating models, using provider contracts and corresponding meter events as input. A rating model describes the pricing scheme for the service, and can be implemented according to a service provider's specific requirements and plugged into the accounting service.

3.7 Profile Service

The Profile Service provides access to user profile information for a user. Basic profile information is collected and supplied by this service, including name, address, user id, to mention a few. In time, this may expand to include more information. The service requestor saves and gets profile information using the Profile Service. The profile key provided by the Profile Service is used by the Contract Service to determine what users have valid contracts with a service provider. In this context, all users of business services must have a profile assigned by the Profile Service. Profiles may be created in advance by using the framework for Web Services Procurement.

To illustrate the functionality of our implementation, we describe next a scenario for Web services procurement that integrates services requestors and providers that has already been implemented.

4 Case of Study

The case of study describes a basic scenario which involves simple accounting information associated with a single web service.
Suppose the following scenario:

1. A services provider, who is a book seller (like Amazon) brings access to its database by means Web services interfaces. The functionalities provided by these interfaces are to get the stock quote, price, availability and other technical features of its products.
2. A client wants to create a virtual enterprise through a services provider. The service provider should offer their product catalogs to the client for the development of the virtual enterprise.

In this scenario, how can client find to the service provider and establish a usage contract with him to carry out the development of the virtual enterprise?

To solve this issue is necessary to use our framework. Firstly, is necessary register the Web services interfaces provided by the services provider within Service Hub. For doing this, our framework present a set of graphic interfaces where new services can be added to view information about the types of services that have already been defined. Under our framework, if a client wants to add her own application there are just a few simple steps to follow:

1. Deploy the service along with an interface WSDL document that defines the service interface and an implementation WSDL for the new service. The WSDL documents must be accessable through a URL.
2. Create an HTML page (e.g. JSP or servlet) that can be used to access and invoke the service. This HTML page must accept the following parameters:
   a. **wsdl** - a URL to the WSDL document describing the service
   b. **namesapce** - the targetNamespace to use in the WSDL document
   c. **servicename** - the service name to use in the WSDL document
   d. **portname** - the port name to use in the WSDL document

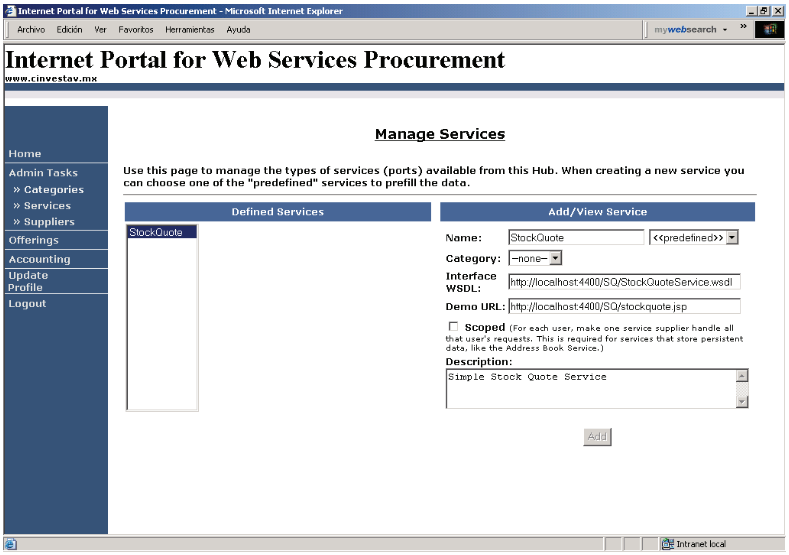
In a similar way, we need to register at least one service provider which provides an implementation of this service. In Fig. 3, a screenshot to add new services in the Service Hub is shown. Once the service and service provider have been registered, is necessary to establish an offering for this service. In this sense, the service provider must create a service offering -- fixed packages including service operations, associated service levels, penalty upon violation, as well as price for using this service -- expressed as a SLA template. A screenshot of our framework where offerings are created is shown in Fig. 4. Next, the client (a service requestor) in order to use this service she must first create a Usage Contract agreeing to the terms and conditions specified in it. Then, she must subscribe to a selected service offering creating a new SLA. The service provider may provide some customization flexibility in its offerings. The customization capability may range from mere selection of a few SLA parameter values (e.g., from a set of fixed throughput levels) as expressed in an offer, to some negotiation of parameters (e.g., negotiation of price for a customer-specified throughput level) to composing new service level objectives. To provide this flexibility, a provider should not only have the required capability of online negotiation, but also its business ability to support any new customer-specified service level objectives (SLOs), i.e., runtime infrastructure for supporting this service level as well as its ability to price this new service level. Therefore, before accepting a new SLA, the provider must ensure its ability to support this new SLA. In Fig. 5, a screenshot to create usage contract is shown. Once the usage contract is established, the client can invoke the service by using our framework. The framework provides Web services dynamic invocation by creating GUIs for consuming the service. The process of Web services invocation is carries out by analyzing WSDL documents. WSDL documents employ XML Schema for the specification of information items either product technical information or business processes operations. Our framework reports the business processes operations, input and output parameters, and their data types in a XML DOM tree which is a XML document. This XML document is presented in HTML format using the Extensible Style-sheet Language (XSL). In Fig. 6, a screenshot for Web services invocation is shown. In this figure, the stock price is displayed given a product code. Each time this Web service is invoked by a service requestor, our framework gathers data which contain accounting information for this usage contract. Among the gathered data are: (1) invocation date, (2) basic price, (3) unit price normal and (4) unit price reduced of usage. With these data, our framework can generate a table where the accounting and billing details for the selected user contract are listed. A screenshot for accounting and billing information is shown in Fig. 7.

Through our framework, the client could find a service provider who offers their product catalogs and provides operations to get the stock quote, price, availability and other technical features of its products. By means of our framework, the client established a usage contract with the service provider and obtained the accounting/bill generation for this service.

As could be observed, our framework demonstrates the various roles and steps that make up the lifecycle of managing and hosting a Web Service, starting with offering up the service to potential requestors, using the service and finally ending with the accounting/bill generation for specific services.

5 Future Directions

As future work, we are considering include management for composite Web services. A composite Web service is one that uses other web services in addition to its own business logic, to fulfill its clients' requests. The underlying Web services may also be composite, thus leading to complex chains of service-to-service interactions. Each service in the composition hierarchy may be independently owned and operated, so that each request must be metered and charged to its immediate client. In addition, it may contribute towards the accounting for a higher-level request in the hierarchy. Thus, the accounting process must include correlation and aggregation of metering data. In a composite service scenario, each web service may potentially be hosted on a separate Service Hub, resulting in a distributed deployment. A request to one service may result in multiple requests to other services on different Service Hubs. The service usage reported by each such *child* request needs to be correlated with the *parent* request that caused it, and aggregated together to compute the composite usage of the *parent* request. Since each service is potentially a composite service, each *child* request may itself act as the parent of other *child* requests. Thus, the correlation and aggregation must be performed recursively, to cover the entire web service call graph generated by an end-user request.



Internet Portal for Web Services Procurement

www.cinvestav.mx

Manage Services

Use this page to manage the types of services (ports) available from this Hub. When creating a new service you can choose one of the "predefined" services to prefill the data.

Defined Services

StockQuote

Add/View Service

Name: StockQuote <<predefined>>

Category: –none–

Interface WSDL: http://localhost:4400/SQ/StockQuoteService.wsdl

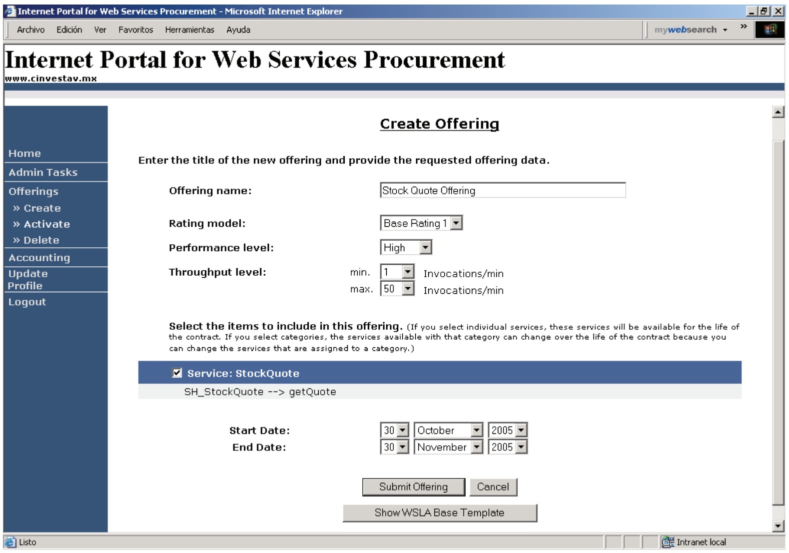
Demo URL: http://localhost:4400/SQ/stockquote.jsp

□ Scoped (For each user, make one service supplier handle all that user's requests. This is required for services that store persistent data, like the Address Book Service.)

Description:

Simple Stock Quote Service

Fig. 3 Graphic interface to add new Web services interfaces within Service Hub



Internet Portal for Web Services Procurement

www.cinvestav.mx

Create Offering

Enter the title of the new offering and provide the requested offering data.

Offering name: Stock Quote Offering

Rating model: Base Rating 1

Performance level: High

Throughput level: min. 1 Invocations/min
max. 50 Invocations/min

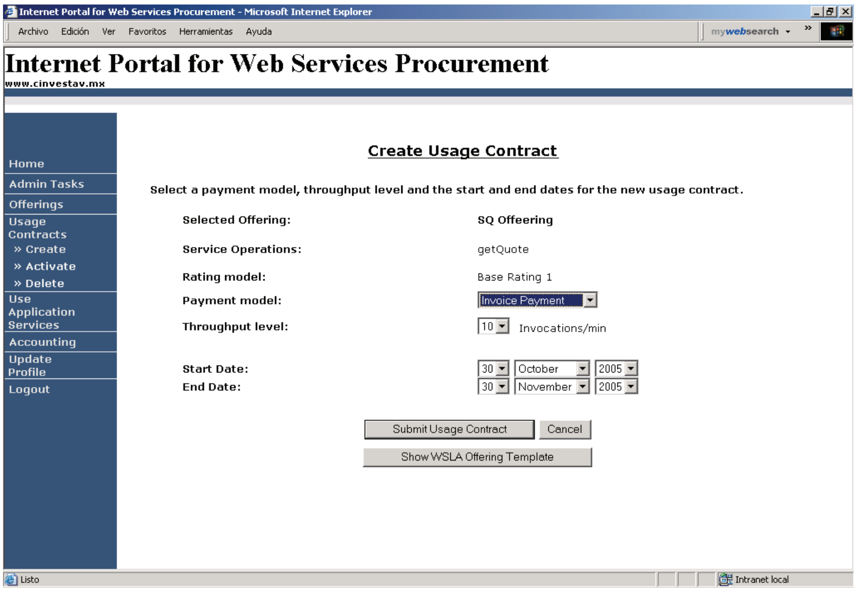
Select the items to include in this offering. (If you select individual services, these services will be available for the life of the contract. If you select categories, the services available with that category can change over the life of the contract because you can change the services that are assigned to a category.)

☑ Service: StockQuote

SH_StockQuote --> getQuote

Start Date: 30 October 2005
End Date: 30 November 2005

Fig. 4 Graphic Interface where offerings can be created by services providers



Internet Portal for Web Services Procurement

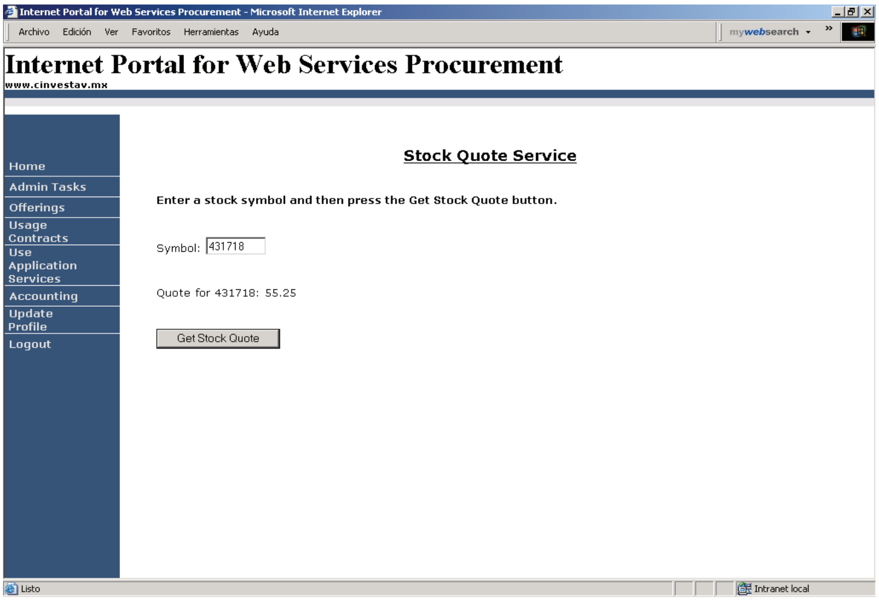
www.cinvestav.mx

Create Usage Contract

Select a payment model, throughput level and the start and end dates for the new usage contract.

Selected Offering:	SQ Offeering
Service Operations:	getQuote
Rating model:	Base Rating 1
Payment model:	Invoice Payment
Throughput level:	10 Invocations/min
Start Date:	30 October 2005
End Date:	30 November 2005

Fig. 5 Graphic Interface where usage contract can be created by services requestors



Internet Portal for Web Services Procurement

www.cinvestav.mx

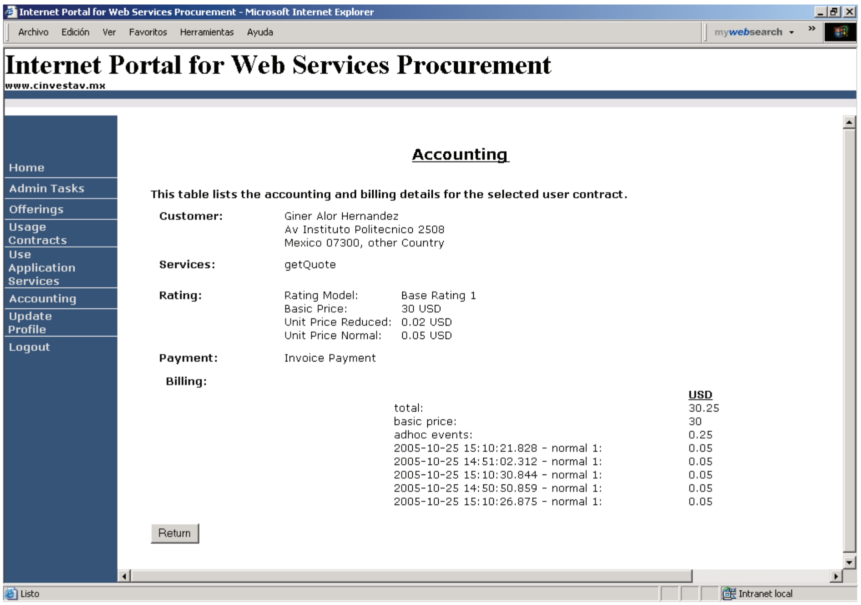
Stock Quote Service

Enter a stock symbol and then press the Get Stock Quote button.

Symbol: 431718

Quote for 431718: 55.25

Fig. 6 Graphic Interface for invoking the Web service that get the stock quote given a product code



Internet Portal for Web Services Procurement

www.cinvestav.mx

Accounting

This table lists the accounting and billing details for the selected user contract.

Customer: Giner Alor Hernandez
Av Instituto Politecnico 2508
Mexico 07300, other Country

Services: getQuote

Rating:		
	Rating Model:	Base Rating 1
	Basic Price:	30 USD
	Unit Price Reduced:	0.02 USD
	Unit Price Normal:	0.05 USD

Payment: Invoice Payment

Billing:

	USD
total:	30.25
basic price:	30
adhoc events:	0.25
2005-10-25 15:10:21.828 - normal 1:	0.05
2005-10-25 14:51:02.312 - normal 1:	0.05
2005-10-25 15:10:30.844 - normal 1:	0.05
2005-10-25 14:50:50.859 - normal 1:	0.05
2005-10-25 15:10:26.875 - normal 1:	0.05

Fig. 7 Graphic Interface for accounting the Web service that get the stock price

6 Related Works

In [9] a classification and comparison framework for Web Services Procurement platforms is presented. This framework is used to compare several existing platforms and to identify some key properties and deficiencies. Furthermore, they present a brief comparison of some quality-aware approaches among different frameworks for Web Services Procurement. The use of cooperative service requester agents within the "alternative offers" protocol is proposed in [10]. Under this idea, service requester agents approaching their pre-set negotiation deadline yet having failed to secure even a single offer would issue a call for help, to which other requester agents will respond by donating their superfluous offers. For doing this, they propose a formal model of this protocol and investigate its effect on improving the success rates in procuring web services. In [11], an implementation issues on a quality-aware approach to Web Services Procurement is presented. The proposed solution is mainly based on using mathematical constraints to define quality-of-service in demands and offers. They developed a prototype of the run-time framework for management and execution of multi-organizational web-based systems. This prototype includes a quality trader web service as the main component, which offers services such as checking for consistency and conformance, and searching for the best choice. Among the main characteristics of implementation, they used the QRL language to specify quality-of-service, and XML to specify QRL-based documents, the definition of XSLT transformations to get the appropriate CSP for carrying out the WSP-related tasks, and the use of a constraint solver as ILOG's OPL Studio. An extension to the user interface functionality to the dynamic SLA negotiation between a customer and several Internet Service Providers is proposed in [12]. For doing this, they implemented an intelligent user interface which is called NIA (Network Interface Agent). The NIA is a multi-agent system which is installed on the user terminal and which is used for the dynamic negotiation of SLS. The SLA is previously established with the Internet Service Provider and specifies that the SLS parameters are dynamically negotiate. The NIA determines the SLS on behalf the user according to the application requirements and the user's needs.

Under the Computing Grid context, some works have been developed in this address. In [13], an architecture and toolkit named GRUBER for resource usage service level agreement (SLA) specification and enforcement in a grid environment is presented. The novelty of GRUBER consists in its capability to provide a means for automated agents to select available resources from virtual organization level on down. It focuses on computing resources such as computers, storage, and networks; owners may be either individual scientists or sites; and virtual organizations are collaborative groups, such as scientific collaborations. A virtual organization is a group of participants who seek to share resources for some common purpose.

7 Conclusions

In this work we have presented a framework for Web services procurement. Our framework features provisioning services, such as contracting, metering, accounting, notification and SLA based management of web services. By means of our framework, service providers can efficiently and flexibly manage their resources to optimize customer satisfaction and, potentially, yield. Furthermore, our framework demonstrates the various roles and steps that make up the lifecycle of managing and hosting a Web Service - starting with offering up the service to potential requestors, using the service and finally ending with the accounting/bill generation for specific services

References

1. Steve Vinoski. Integration with Web Services. IEEE Internet Computing. November-December 2003 pp 75-77.
2. Adams, H., Dan Gisolfi, James Snell, Raghu Varadan. "Custom Extended Enterprise Exposed Business Services Application Pattern Scenario," http://www-106.ibm.com/developerworks /webservices/library/ws-best5/, Jan. 1, 2003
3. Samtani, G. and D. Sadhwani, "Enterprise Application Integration and Web Services," in Web Services Business Strategies and Architectures, P. Fletcher and M. Waterhouse, Eds. Birmingham, UK: Expert Press, LTD, pp. 39-54, 2002a.
4. Alexander Keller, Heiko Ludwig: Defining and Monitoring Service Level Agreements for dynamic e-Business. In Proceedings of the 16th USENIX System Administration Conference (LISA'02), November, 2002.

5. Heiko Ludwig, Alexander Keller, Asit Dan, Richard P. King. A Service Level Agreement Language for Dynamic Electronic Services. In Proceedings of WECWIS 2002, Newport Beach, CA, pp. 25 - 32, IEEE Computer Society, Los Alamitos, 2002.
6. Alexander Keller, Gautam Kar, Heiko Ludwig, Asit Dan, Joseph L. Hellerstein. Managing Dynamic Services: A Contract Based Approach to a Conceptual Architecture. IBM Research Technical Report RC22162, 2002.
7. HP OpenView Service Desk 4.5. Release Notes. First Edition. July 2002. Hewlett-Packard Company. 3000 Hanover Street. Palo Alto, CA 94304 U.S.A.
8. Frank Leymann. Web Services: Distributed Applications without Limits - An Outline. IBM Software Group. June 25, 2003.
9. Octavio Martín-Díaz, Antonio Ruiz-Cortés, Rafael Corchuelo, Miguel Toro. A Framework for Classifying and Comparing Web Services Procurement Platforms. Proceedings of the Fourth International Conference on Web Information Systems Engineering Workshops (WISEW'03).
10. A.M.Abdoessalam and N.Mehandjiev. Collaborative Negotiation in Web Service Procurement. Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'04).
11. Octavio Martín-Díaz, Antonio Ruiz-Cortés, David Benavides, Amador Duran, and Miguel Toro. A Quality-Aware Approach to Web Services Procurement. In B. Benatallah and M.-C. Shan (Eds.): TES 2003, Lecture Notes on Computer Science 2819, pp. 42–53, 2003.
12. Gilles Klein and Francine Krief. Mobile Agents for Dynamic SLA Negotiation. In E. Horlait (Ed.): MATA 2003, Lecture Notes on Computer Science 2881, pp.23 –31, 2003.
13. Catalin L. Dumitrescu and Ian Foster. GRUBER: A Grid Resource Usage SLA Broker. In J.C. Cunha and P.D. Medeiros (Eds.): Euro-Par 2005, Lecture Notes on Computer Science 3648, pp. 465–474, 2005.

High Level Parallel Compositions (CPANs) for the Parallel Programming based on the use of Communication Patterns

Mario Rossainz López[1], Manuel I. Capel Tuñón[2]

[1] Benemérita Universidad Autónoma de Puebla, Avenida San Claudio y 14 Sur,
San Manuel, Puebla, State of Puebla, 72000, México
mariorl@siu.buap.mx
http://www.cs.buap.mx/~mrossainz
[2] Departamento de Lenguajes y Sistemas Informáticos, ETS Ingeniería Informática,
Universidad de Granada, Periodista Daniel Saucedo Aranda s/n,
18071, Granada, Spain
mcapel@ugr.es
http://lsi.ugr.es/~mcapel

Abstract. This article presents a programming methodology based on High Level Parallel Compositions (CPAN in the Spanish acronym) within a methodological infrastructure made up of an environment of Parallel Objects [10], an approach to Structured Parallel Programming and the Object-Orientation paradigm. The implementation of commonly used communication patterns is explained by applying the method (the CpanFarm, CpanPipe and CpanTreeDV that represent respectively, the patterns of communication Farm, Pipeline and Binary Tree, the latter one used within a parallel version of the design technique known as Divide & Conquer), which conforms a library of classes suitable for use in applications within the programming environment of the C++ and POSIX standards for thread programming. Thus, in this work presents the design of the CPAN that implements a parallelization of the algorithmic design technique named Branch & Bound and uses it to solve the Travelling Salesman Problem (TSP).

1 Introduction

Obtaining efficiency in parallel programs is not so much a problem of acquiring processor speed, but rather, it is about how to program efficient interaction/communication patterns among the processes [1], [2], [4], [6] to achieve the maximum possible speed-up of a given parallel application. Parallel Programming based on the use of communication patterns is known as Structured Parallel Programming (SPP) [6], [7]. The widespread adoption of SPP methods by programmers and system analysts currently presents a series of open problems. We are particularly interested in proposing new solutions to the following: (a) the lack of SPP methods applicable to the development of a wider range of software applications; (b) the determination of a complete set of communication patterns and their semantics; (c) the necessity to make predefined communication patterns or high level parallel compositions available to the community, aimed at encapsulating parallel code within programs; (d) the adoption of a sound (i.e. without *anomalies*) programming approach based on merging concurrent primitives and Object-Oriented (O-O) features, thereby meeting the requirements of *uniformity, genericity* and *reusability* of software components [6]. The present investigation is focused on SPP methods, and a new implementation is proposed (carried out with C++ and the POSIX Threads Library) of a library of High Level Parallel Composition (CPAN) [6], [7] classes, which provide the programmer with the communication patterns most commonly used in Parallel Programming. At the moment, the library includes the following ones: CpanFarm, CpanPipe, CpnaTreeDV, the latter one being used in a parallel version of Divide & Conquer algorithmic design technique and CpanFarmBB that is one pattern composed with Farm process that implements a parallelization of the algorithmic design technique named Branch & Bound.

1.1 The Problem Being Tackled

In order to cope with the above described items, we have found that an O-O Parallel Programming environment providing the features listed below must be used, (a) capacity of object method invocation that assumes asynchronous message passing and asynchronous futures; (b) the objects should have internal parallelism; (c) availability of different communication mechanisms when service of petitions from client processes take place in parallel; (d) distribution transparency of processes within parallel applications; (e) Programmability, portability and performance, as a consequence of software development within an O-O programming system.

1.2 Scientific Objectives in this Research

The current investigation has mostly been carried out within the PhD thesis research work referenced in [8], whose achieved operational objectives are listed below:

1. To develop a programming method based on High Level Parallel Compositions or CPANs.
2. To develop a library of classes of parallel objects [10] that provides the programmer or the analyst with a set of commonly used communication patterns for parallel programming; the objects should be uniformly programmed as reusable, generic, CPANs.

To offer this library to the programmer, so that he/she can exploit it by defining new patterns, adapted to the communication structure of processes in his/her parallel applications, by following an O-O programming paradigm, which includes class inheritance and object generic instantiation as its main reusability mechanisms.

2 High Level Parallel Compositions or CPANs

The basic idea of the programming method consists of the implementation of any type of communication patterns between parallel processes of an application or distributed/parallel algorithm as CPAN classes, following the O-O paradigm. CPANs are aimed at helping parallel applications programmers in programming efficient, portable and easy to program code by encapsulating parallelism or communication protocols from the sequential application processes of the parallel applications [8]. CPANs are structured as three classes of parallel objects [10], see Fig 1:

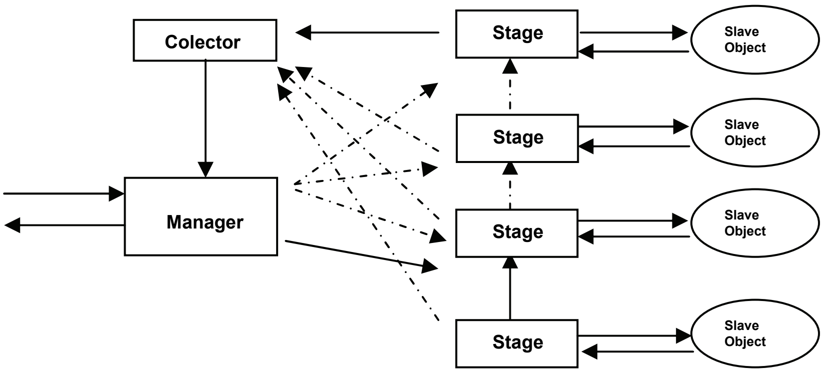


Fig. 1. Internal Structure of a CPAN

An object manager, which is the only visible interface to the sequential processes in a parallel application, composed of the collector and stages objects and should be coordinated by the manager itself, see Fig 2.

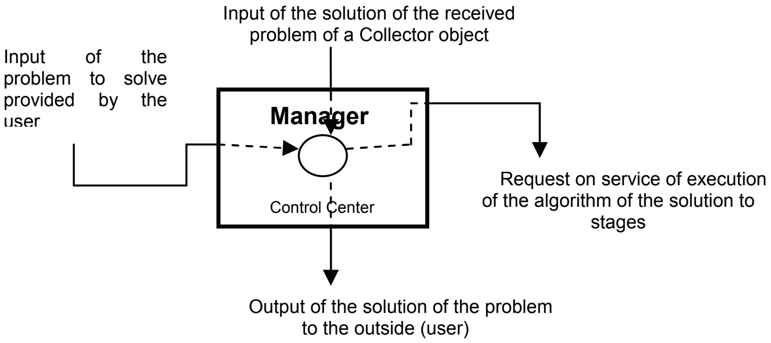

Input of the solution of the received problem of a Collector object
Input of the problem to solve provided by the user
Manager
Control Center
Request on service of execution of the algorithm of the solution to stages
Output of the solution of the problem to the outside (user)

Fig. 2. The Manager Object (Internal Structure)

The stage objects intended to configure a connection topology among these objects in order to provide a given communication pattern semantics. The stage objects are objects of specific purpose responsible for encapsulating a client-server type interface between the manager and the object slaves (objects that are not actively participative in the composition of the CPAN, but rather, are considered external entities that contain the sequential algorithm constituting the solution of a given problem), see Fig 3.

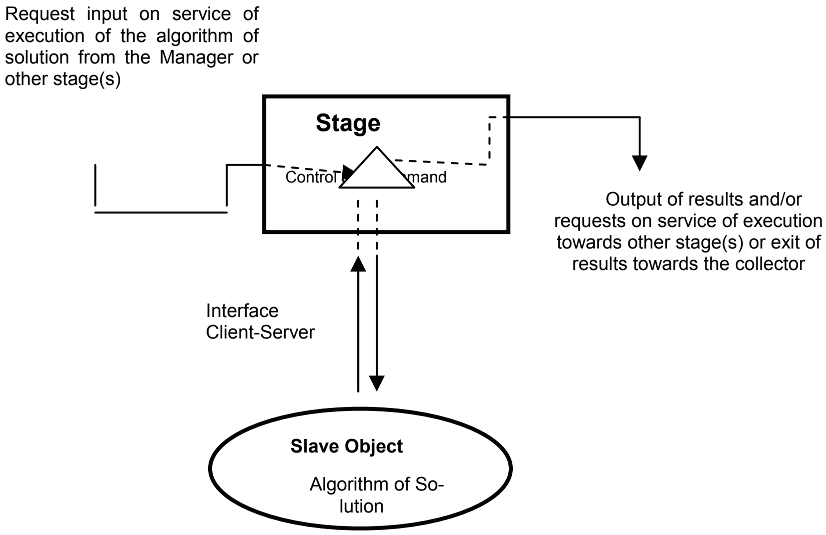

Request input on service of execution of the algorithm of solution from the Manager or other stage(s)
Stage
Control
mand
Output of results and/or requests on service of execution towards other stage(s) or exit of results towards the collector
Interface Client-Server
Slave Object
Algorithm of Solution

Fig. 3. The Stage Object (Internal Structure)

An object collector in charge of storing in parallel the results received from the stages during the service of a sequential process petition. The control flow within the stages of a CPAN depends on the communication pattern implemented between these. When the CPAN concludes its execution, the result does not return to the manager directly, but rather to an instance of the class Collector, which takes charge of storing these results and of sending them to the manager, which then sends them to the exterior as they arrive, i.e., without begin necessary to wait for all the results to be obtained at the end of the computation. See Fig 4.

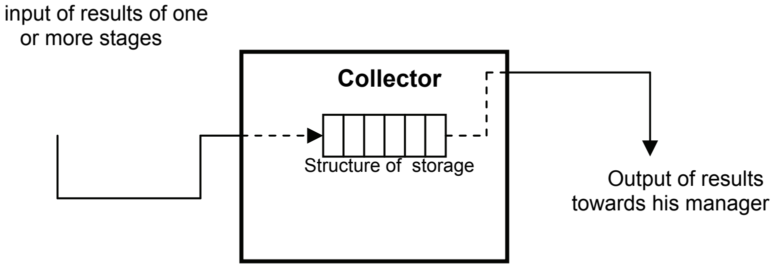

input of results of one or more stages
Collector
Structure of storage
Output of results towards his manager

Fig. 4. The Collector Object (Internal Structure)

2.1 Types of Communication Between the Parallel Objects

1. *The synchronous way* stops the client's activity until the object's active server gives back the answer to the petition.
2. *The asynchronous way* does not force any waiting in the client's activity; the client simply sends its petition to the active server and then it continues.
3. *The asynchronous future way* makes only to wait the client's activity when the result of the invoked method is needed to evaluate an expression during its code execution.

2.2 Basic classes of a CPAN

The abstract class ComponentManager defines the generic structure of the component manager of a CPAN, from which all the concrete manager classes are derived, depending on the parallel behavior which is needed to create a specific CPAN.

The abstract class ComponentStage defines the generic structure of the component stage of a CPAN as well as its interconnections, so that all the concrete stages needed to provide a CPAN with a given parallel behavior can be obtained by class instantiation.

The concrete class ComponentCollector defines the concrete structure of the component collector of any CPAN. It implements a multi-item buffer, which permits the storage of the results from stages that make reference to this collector.

2.3 The Synchronization Restrictions MaxPar, Mutex and Sync

Synchronization mechanisms are needed when several petitions of service take place in parallel in a CPAN, being capable its constituting parallel objects of interleaving their concurrent executions while, and at the same time, they preserve the consistency of the data being processed [10]. Within the code of any CPAN, execution constraints are automatically included when the methods MaxPar (Maximum Parallelism), MutEx (Mutual Exclusion) and Sync (Synchronization type producer-consumer) of the library are called. The latter ones must be used to obtain a correct programming of object methods and to guarantee data consistency in applications.

3 The CPANs Farm, Pipe and TreeDV

The parallel patterns applied until now have been the *Pipeline*, the *farm* and the *treeDV*.

The Pipeline is made up of a set of interconnected stages, one after another, in which the information flows between these until an ending condition is determined in one of them. At this moment the pipeline enters in another execution mode in which each stage unloads its data to the next one. The last stage is responsible for sending the processes data to the Collector. See Fig 5.

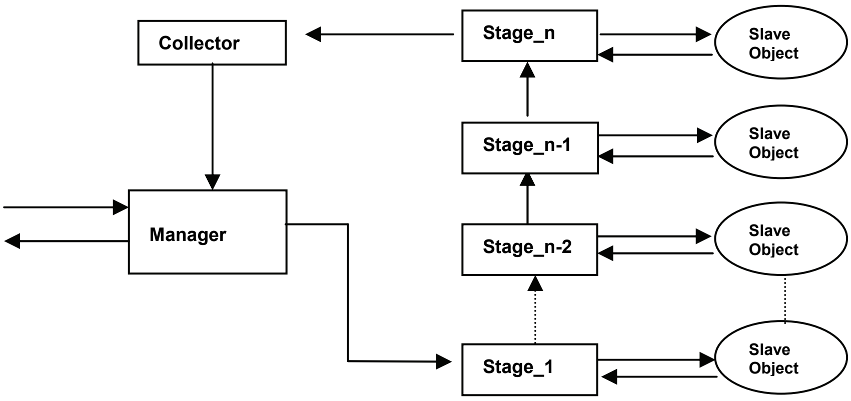


Fig. 5. The CPAN of a Pipeline

The *Farm* is composed of a set of worker processes executed in parallel until a common objective is reached, and a controller in charge of distributing work and controlling the progress of the global calculation. See Fig 6.

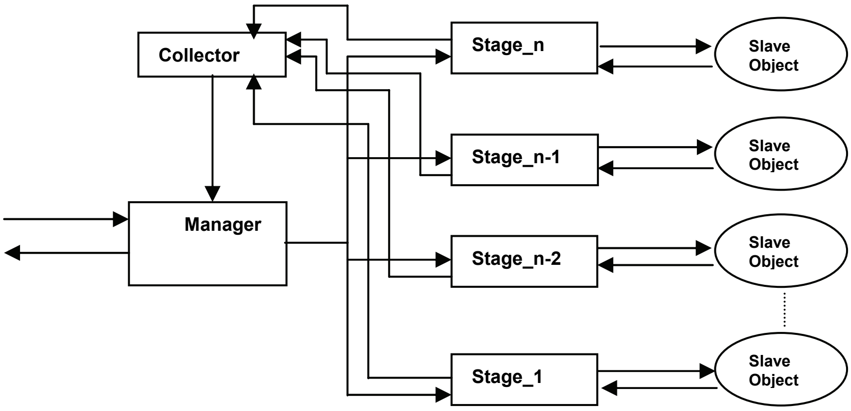


Fig. 6. The CPAN of a Farm

The *TreeDV* is a communication pattern in which the information flows from the root to the leaves of the tree and vice versa. The nodes on the same level are executed in parallel in order to implement a parallel version of the so called Divide & Conquer algorithmic design technique. The stage situated at the root of the TreeDV will obtain the solution of the problem when the global calculation finishes. This CPAN is configured in a similar way. See Fig 7.

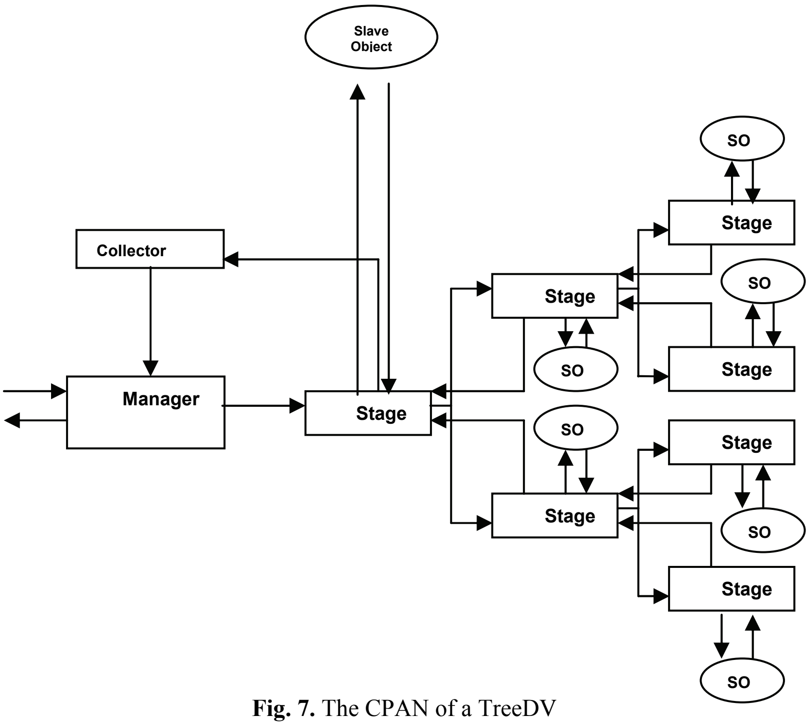


Fig. 7. The CPAN of a TreeDV

These constitute a significant set of reusable communication patterns in multiple parallel applications and algorithms. See [5] and [8] for details.

3.1 Results Obtained

Some CPANs adapt better to the communication structure of a given algorithm than others, therefore yielding different speedups of the whole parallel application. The way in which it must be used to build a complete parallel application is detailed below.

1. It is necessary to create an instance of the adequate class manager, that is to say, a specialized instance (this involves the use of inheritance and generic instantiation) implementing the required parallel behavior of the final manager object. This is performed by following the steps:
   1.1. Instance initialization from the class manager, including the information, given as associations of pairs (slave_obj, associated_method); the first element is a reference to the slave object being controlled by each stage and the second one is the name of its callable method.
   1.2. The internal stages are created (by using the operation *init()*) and, for each one, the association (slave_obj, associated_method) is passed to. The second element is needed to invoke the associated_method on the slave object.

2. The user asks the manager to start a calculation by invoking the *execution()* method of a given CPAN. This execution is carried out as it follows:
   2.1. a collector object is created for satisfying this petition;
   2.2. input data are passed to the stages (without any verification of types) and a reference to the collector;
   2.3. results are obtained from the object collector;
   2.4. The collector returns the results to the exterior without type verification.
3. An object manager will have been created and initialized and some execution petitions can then start to be dispatched in parallel.

We carried out a Speedup analysis of the Farm, Pipe and TreeDV CPANs for several algorithms in an Origin 2000 Silicon Graphics Parallel System (with 64 processors) located at the European Center for Parallelism in Barcelona (Spain), this analysis is discussed below.

Assuming that we want to sort an array of data, some CPANs will adapt better to communication structure of a Quicksort algorithm than others. These different parallel implementations of the same sequential algorithm will therefore yield different speedups. The program is structured of six set of classes instantiated from the CPANs in the library High Level Parallel Compositions, which constitute the implementation of the parallel patterns named Farm, Pipe and TreeDV. The sets of classes are listed below:

1. *The set of the classes base*, necessary to build a given CPAN.
2. The set of the classes that define the abstract data types needed in the sorting.
3. *The set of classes that define the slave objects*, which will be generically instantiated before being used by the CPANs.
4. The set of classes that define the Cpan Farm.
5. The set of classes that define the Cpan Pipe.
6. The set of classes that define the Cpan TreeDV.

This analysis of speedup of the CPANs appears in Figures 8, 9 and 10. In all cases the implementation and test of the CPANs Farm, Pipe and TreeDV 50000 integer numbers were randomly generated to load each CPAN.

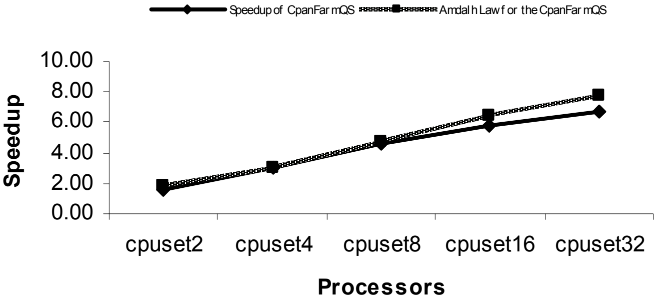


Fig. 8. Scalability of the Speedup found for the CpanFarm in 2, 4, 8, 16 and 32 processors

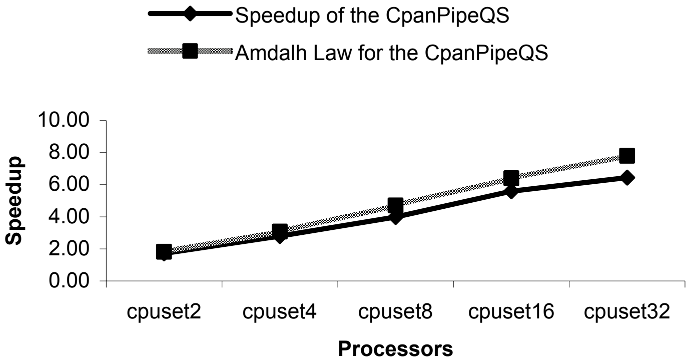


Fig. 9. Scalability of the Speedup found for the CpanPipe in 2, 4, 8, 16 and 32 processors

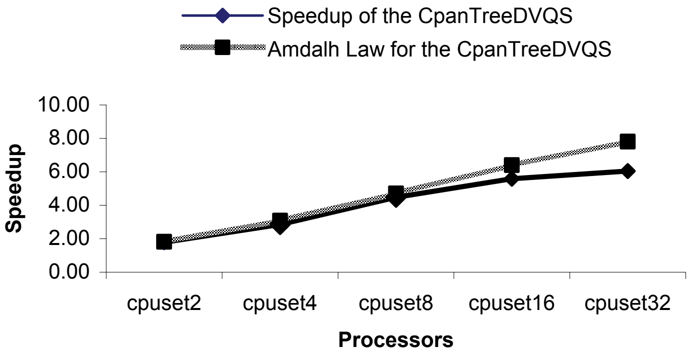


Fig. 10. Scalability of the Speedup found for the CpanTreeDV in 2,4,8, 16 and 32 processors

4 Parallelization of the Branch & Bound Technique

Branch-and-bound (BB) makes a partition of the solution space of a given optimization problem. The entire space is represented by the corresponding BB *expansion tree*, whose root is associated to the initially unsolved problem. The children nodes at each node represent the subspaces obtained by *branching,* i.e. subdividing, the solution space represented by the parent node. The leaves of the BB tree represent nodes that cannot be subdivided any further, thus providing a final value of the cost function associated to a possible solution of the problem.

Three stages are performed during the execution of a program based on a BB algorithm:

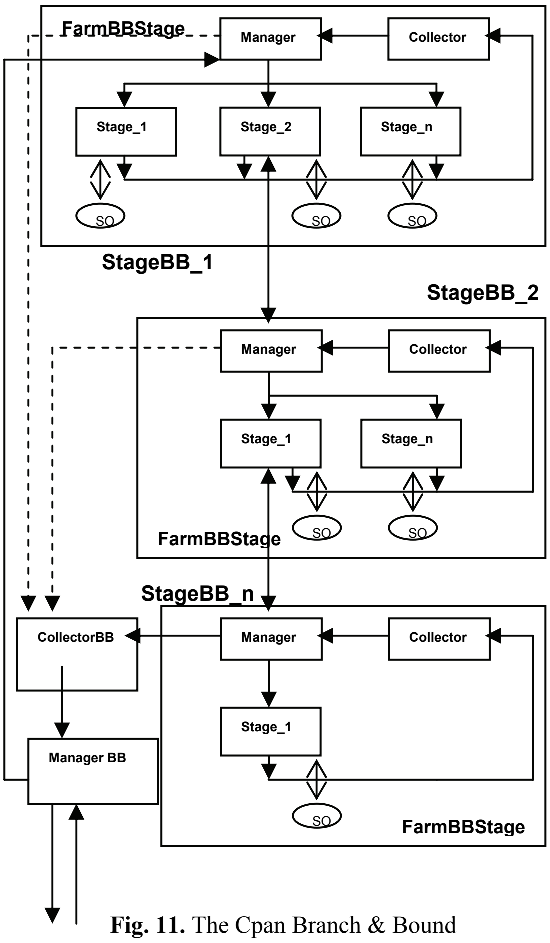


Fig. 11. The Cpan Branch & Bound

1. *Branch:* the node selected in the previous step is subdivided in its children nodes by following a ramification scheme to form the expansion tree. Each child receives from its father node enough information to enable it to search a suboptimal solution.
2. *Bound:* Some of the nodes created in the previous stage are deleted, i.e. those whose partial cost, which is given by the cost function associated to this BB algorithm instance, is greater than the best minimum bound calculated up to that point.

The ramification is generally separated from the bounding of nodes on the expansion tree in parallel BB implementations, and so we followed this approach using *a Farm communication scheme* [9]. The expansion tree, for a given instance of the BB algorithm, is obtained by iteratively subdividing the stage objects according to this pattern until a stage representing a leaf-node of the expansion tree is found, see Fig 11.

The pruning is implicitly carried out within another *farm* construction by using a *totally connected scheme* between all the processes. The manager can therefore communicate a sub-optimal bound found by a process to the rest of the branching processes and thus avoid unnecessary ramifications of sub-problems. *The Cpan Branch & Bound* is composed of a set of *Cpans Farm*; see Figure 11, which represent each one a set of worker processes and one manager, therefore, forming a new type of structured Farm, the *Farm Branch & Bound or FarmBB*, which is also included in the library of CPANs. All the worker processes of the *Farm BB* are executed in parallel, thereby forming the expansion tree of nodes given by the BB algorithm technique. The initial problem, or the root of the expansion tree, is given to the manager process of the initial *Cpan Farm*, which is in charge of distributing the work and of controlling the global calculation progress. It is also responsible for sending results to the collector of the *Cpan FarmBB*, which will display them [9].

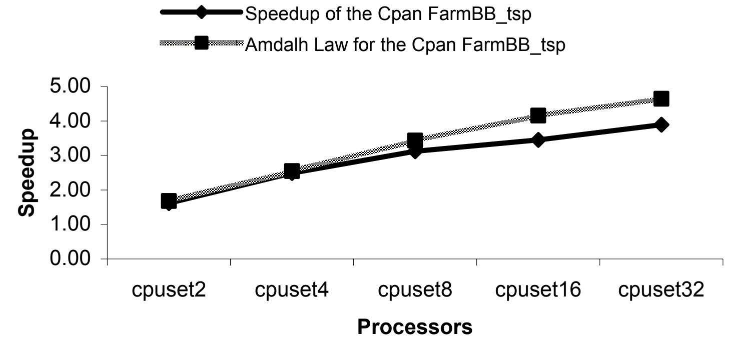


Fig. 12. Speedup of parallel CpanBB with N=50 cities in 2, 4, 8, 16 and 32 processors

The CPAN based parallel BB algorithm was tested by solving the TSP with 50 cities and by using the first best search strategy driven by a *least cost* function associated to each live node. The results obtained yielded a deviation ranging from 2% (2 processors) to 16% (32 processors) with respect to the optimal ones, as predicted by the Amdalh law for this parallelized algorithm. See Fig 12.

5 Conclusions

The programming method presented is based on Corradi's High Level Parallel Compositions, but updated and adapted to be used with the C++ programming language and POSIX standard for thread programming. The CPANs Pipe, Farm, and TreeDV comprise the first version of a library of classes intended to be applied to solve complex problems such as the afore-mentioned parallelization of the Branch & Bound technique, thus offering an optimal solution to the TSP NP-Complete problem.

References

1. Brinch Hansen; “Model Programs for Computational Science: A programming methodology for multicomputers”, Concurrency: Practice and Experience, Volume 5, Number 5, 407-423, 1993.
2. Brinch Hansen; “SuperPascal- a publication language for parallel scientific computing”, Concurrency: Practice and Experience, Volume 6, Number 5, 461-483, 1994.
3. Capel M.I., Palma A., “A Programming tool for Distributed Implementation of Branch-and-Bound Algorithms”. Parallel Computing and Transputer Applications. IOS Press/CIMNE. Barcelona 1992.
4. Capel, M.; Troya J. M. “An Object-Based Tool and Methodological Approach for Distributed Programming”. Software Concepts and Tools, 15, pp. 177-195. 1994.
5. Capel, M.; Rossainz, M. “A parallel programming methodology based on high level parallel compositions”. Proceedings of the 14th International Conference on Electronics, Communications and Computers, 2004, IEEE CS press. 0-7695-2074-X.
6. Corradi A, Leonardo L, Zambonelli F. “Experiences toward an Object-Oriented Approach to Structured Parallel Programming”. DEIS technical report no. DEIS-LIA-95-007. 1995
7. Danelutto, M.; Orlando, S; et al. “Parallel Programming Models Based on Restricted Computation Structure Approach”. Technical Report-Dpt. Informatica. Universitá de Pisa.
8. Rossainz, M. “Una Metodología de Programación Basada en Composiciones Paralelas de Alto Nivel (CPANs)”, Universidad de Granada, PhD dissertation, 02/25/2005.
9. Rossainz M, Capel M. “Design and use of the CPAN Branch & Bound for the solution of the traveling salesman problem (TSP)”. Proceedings of the ECMS 2005 – HPC&S. Riga Latvia, 2005. ISBN: 1-84233-113-2.
10. Rossainz M, Capel M. “An Approach to Structured Parallel Programming Based on a Composition of Parallel Objects”. Congreso Español de Informática CEDI-2005. XVI Jornadas de Paralelismo. Granada, Spain 2005. Editorial Thomson. ISBN: 84-9732-430-7.

Information Systems

Italianitá: Discovering a Pygmalion effect on Italian Communities Using Data Mining

Alberto Ochoa[1,2], Alán Tcherassi[2], Inna Shingareva[3], A. Padméterakiris[4], J. Gyllenhaale[5] &, José Alberto Hernández[6]

1. Facultad de Ingeniería Eléctrica, Universidad Autónoma de Zacatecas, Av. Ramón López Velarde #801; C.P. 98000 Zacatecas, México,
2. Computer Institute (Postdoctorale Program), State University of Campinas; Postal Box 6176, 13084-971 Radamaelli – SP, Brazil.
3. Artificial Intelligence Institute, Kazaksthan University; Astana, Kazaksthán.
4. Larissa University; Larissa, Grecia.
5. Manx University; Ramsey, Man Island.
6. Centro de Investigación en Ingeniería y Ciencias Aplicadas, Universidad Autónoma del Estado de Morelos; México

cbr_lad7@yahoo.com.mx [1]

Abstract. The present paper discusses an investigation related to the Social Data Mining field using WEKA, a tool that mine information of the structure and content of activities made by descendants of Italians with the purpose of discovering a Pygmalion Effect, which consists of a conduct change of a group that shares similar characteristics induced by the expectations of the same one, this phenomena has been documented since the Sixties, but with few detailed research with truly information, for this purpose we applied a questionnaire to people of four Italian communities whose are scholarship holders of the "RAI Internazionale", to explore their daily activities made on the Internet.

Keywords Pygmalion Effect, Data mining, Modeling of societies.

1 Introduction

Social Data Mining Systems allow the analysis of the society's behavior. These systems do that by mining and redistributing the information on computer files storing the social activity like Usenet messages, log files, purchasing records and links of interest. Although, we generate two general questions to evaluate the performance of such systems: (1) is the extracted information of any value? And (2) is possible to determine if a set of physical separated people can show a similar way of thinking about likes and preferences?

We made an analysis that provides positive answers for both questions. First, a number of attributes about web sites give us as a result the prediction of the behavior on the use of specific computer skills.

We live in an age plenty of information. The Internet offers endless possibilities. Web sites to experience, music to listen, chats rooming, and unimaginable products and services offering to the consumer an endless options varying in quality. People are experiencing difficulties to manage the information: they can not and do not have time to evaluate the whole options by themselves, unless the situation seriously forces them to do that.

In sixties decade appears the first serious studies to understand the Pygmalion effect, which try to demonstrate how "normal" people are induced to behave in a different way, when they show pertaining to a particular group.

In this paper we try to describe how four groups of individuals with common ancestors can make computational activities and web purchases in a similar way. A task to manage information which several internet users must do is "the subject management": searching, evaluating and organizing information resources for a specific subject, sometimes Users search for professional interest subjects, some other times just for personnel interest. Users can create information storage collections in the web for personnel use or to share with partners at work or with friends.

Our approach to this problem combines social data mining [20] with information about work spaces [4]. As the cluster of this People in Web [13], follows certain patterns, this can be analyzed by means of these techniques. In the daily life, when people desire forming part of a social group, without having the knowledge to chose among different alternatives, they trust frequently on the experience and opinions of others. They look for advice in their ethnic-social group, familiar with certain likes and ways of thinking. When evaluating the offered perspectives by similar/near persons to them, or from recognized experts on a subject. For instance, a Usenet of users of Italian origin can recommend certain type of food and where to buy the ingredients also, when registers of these activities exist, these can be analyzed. For our research we need this information to understand how these sites on the web are populated and conformed. Social data mining can be applied to analyze the records generated on the web [16] (answering the question: Which are the most visited sites for the most of people?), online conversations [24] (Which are the sites where people purchase "thematic" things or for a community?), or web log files [13] (Which sites are the most visited?). By means of social data mining is taken the final move.

This paper is organized in five sections. In section one, we introduce our paper. En section two, we describe the ethnic-social effect called "Pygmalion Effect", we describe how can be discovered using data mining, we describe an approach named "Social Data Mining" also. In section three we discuss the application of WEKA to confirm the hypothesis of our research. In section four, we discuss the tests made to the analyzed information. In section number five, we discuss the results generated for the tests, and finally on the last section, we give the conclusions of our research.

2 The Pygmalion Effect

By the end of the Sixties, a professor of psychology called Robert Rosenthal, made the following experiment: joined the teachers of a school and showed them a test made among the students, which indicates that some students were more "shining" than others. "Of these students we can wait for great results ", assured to them. In fact - and responding to the objectives of the experiment- that test was simulated by Rosenthal [17], to induce the teachers to think that certain students had more potential that the rest. Nevertheless, after eight months, those students indeed obtained better qualifications than the average of the class. Like teachers believed in "the supposedly shining" students, offered to them more attention, support, time and feedback. This abundance of conditions was soon translated in a better learning and - in better qualifications. Those children did not stand out being intelligent, but because their teachers believed that they were. Through its experiment, Rosenthal discovered that the expectations of the teachers were reflected in the performance of the students. His conclusion was the following one: while higher are the expectations that a person has with respect to other, more probable than this last one obtains positive results. This discovery put in evidence a phenomenon that is known with the name of "The Pygmalion Effect".

2.1 Data Mining

Data Mining, is the extraction of hiding and predictable information inside great data bases, is a powerful new technology with great potential to help to the companies or organizations to focus on the most important information in their Bases of Information (Data Warehouse). Data Mining tools predict future tendencies and behaviors, allowing businesses to make proactive decisions leaded by knowledge-driven information.

The automated prospective analyses offered by a product thus go beyond past events provided by retrospective typical tools of decision support systems. Data Mining tools can respond to questions of businesses that traditionally consume too much time to be solved and to which the users of this information almost are not willing to accept. These tools explore the data bases searching for hidden patterns, finding predictable information that sometimes an expert cannot find because this is outside expectations.

2.2 Justification

Most of the social groups, that immigrate to another country form communities whose share common characteristics.

As time pass, these characteristics are reinforced if the number of members is considerable, or are assimilated by a greater group [9]. Due that the Pygmalion effect is not considered completely an ethnic-social effect, the necessity to propose, the analysis of the information using Data Mining, with study aims, has allowed to discover the "Italianitá" that they thought they had, and how this marked their activities and forms of using the Web.

2.3 Data Mining Applications in Social Aspects

One of the most transcendental aspects of the use of Data mining is denominated Social Data Mining, which tries to find different patterns in predefined clusters in the network, like the groups of discussion, Usenets, thematic chats among others. Other work has been focused on extracting information about online conversations such as the USENET PHOAKS [19] mining messages in the USENET newsgroup that recommend Web sites. Categorizing the users mentions to create lists of popular Web sites for each group. Where? [22] Has been analyzed the newsgroup information and the Usenet conversations and if they have been used to create visualizations of the conversations. These visualizations can be used to find conversations with the desirable characteristics, such as equality of participation or regular participants. In [6] also was extracted information of newsgroups and visualizations of the conversation subject, contributions of individual messages, and the relation among them were designed. Another research has been centered in extracting the information of web user records. The Log files [23] register information of the users, analyze this to find common connections between Web pages, and they construct diverse visualizations of these data to help user navigation through Web sites. Persecuting the navigation metaphor, some investigators have used the term "social navigation" in order to characterize the work of this nature [11]. Finally, a different technical approach [2] uses the register of activity - e.g., a sequence of visited URLs during a session like the basic unit. Based on this, they have developed techniques to calculate similarities between the trajectories of sequences and to make recommendations - for example, to similar pages to the visited ones.

2.3.1 Social Data Mining

The motivation to make an approach by means of applications with Data Mining is based on previous works of Social Data Mining in this research area [3]. This research area emphasizes the role of the collective analysis of conduct effort, rather that the individual one. A social tendency results from the decisions of many individuals, joined only in the location in where they choose to coexist, yet this, still it reflects a rough notion of what the researchers of the area find of what could be a correct and valid social tendency [21]. The social tendency reflects the history of the use of a collective behavior, and serves like base to characterize the behavior of future descendants [8]. The Data Mining approaches for social aspects look for analogous situations in the behavior registers [14]. The investigators look for situations where the groups of people are producing computer registers (such as documents, USENET messages, or Web sites and links to groups with a specific profile) like part of its normal activity. The potentially useful information implicit in these files is identified; and the computer techniques to display the results are designed. Thus the computer discovers and makes explicit the "social tendencies through the time" created by a particular type of community.

The systems that analyze social aspects with Data Mining do not require expert users in no new activity, due to this, the investigators in the subject try to explore the information of the users preference implicit in the existing activity registers.

3 System Development

The system will be able to analyze the behavior for each one of the samples of the Italian Communities, from the information of the RAI Internazionale scholarship holders, by means of WEKA use, which has demonstrated being an efficient tool for searching hiding parameters that must be discovered [18]. The compiled information was analyzed to discover behavior patterns that share these individuals, and based on their gender and age, we determine if this behavior was an innate or induced tendency by their family of Italian origin.

3.1 Methodology

The name of Data Mining derives from the similarities between looking for valuable information in great data bases - for example: to find information of the tendencies of the society behavior in great amounts of stored Gigabytes – and mining a mountain to find a vein of valuable metals. Both processes require to examine an immense amount of material, or to investigate intelligently until finding exactly where the values reside (see Figure 1). If data bases of sufficient size and quality are available, the Data Mining technology can generate new opportunities of interpretation when providing these capacities.

3.1.1 Automatic Prediction of Tendencies and Behavior.

Data mining automates the process to find predictable information in great data bases (See Figure 1). Questions that traditionally required an intensive manual analysis now can be directly and quickly answered from the data [14].

A typical example of a predictable problem is the marketing oriented to objectives (targeted marketing). Data mining uses data derived of previous promotional mailing campaigns to identify possible objectives to maximize results of the investment in future mailing. Other predictable problems include forecasting of future financial problems and other forms of breach, and identify the population segments that respond probably to similar events.

3.1.2 Automatic Discovering of Previously Unknown Models

The Data Mining Data tools sweep the data bases and identify previously hidden models in only one step. Other problems for models discovering include the detection of credit card fraudulent transactions and to identify abnormal data that can represent keypunch errors in the load of data.

The Data mining techniques can generate benefits for the automatization of existing hardware and software platforms and can be implemented into new systems as the existing platforms get updated and new products are developed[7].

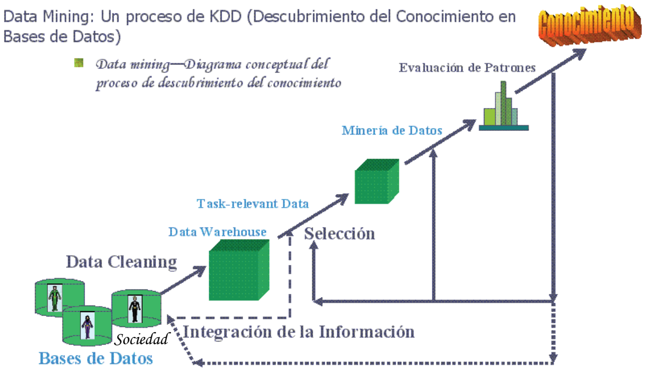

Data Mining: Un proceso de KDD (Descubrimiento del Conocimiento en Bases de Datos)
Data mining—Diagrama conceptual del proceso de descubrimiento del conocimiento
Conocimiento
Evaluación de Patrones
Minería de Datos
Task-relevant Data
Data Warehouse
Selección
Data Cleaning
Sociedad
Integración de la Información
Bases de Datos

Fig. 1. Data Mining process. The society information inside a *Data bases* is cleaned and stored in a *Data Ware House*, then is mined by means of a loop back *selection* and *patterns evaluation* process processes

When the Data mining tools are implemented in high performance parallel processing systems, can analyze massive data bases in just few minutes. Faster processing means that the users can automatically experiment with more models to understand complex data [5]. High speed is practical for the user and makes possible to analyze immense amounts of data. The great data bases, as well, can produce better predictions.

The data bases can be huge as well on depth as well as on width.

More columns. So many times analysts must limit the number of variables to examine when manual analysis are done due limitations on time. However, variables that are suppressed because they seem without importance can provide information about unknown models. A high performance Data mining allows users to explore the whole data base, without a set of variables preselection [10].

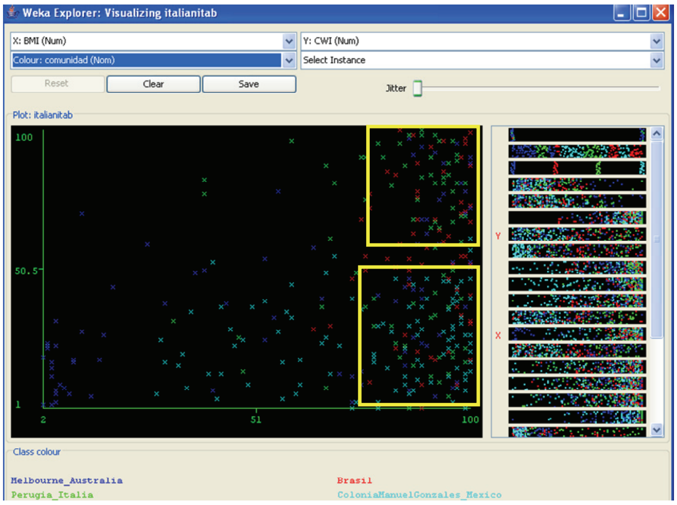
More rows. Bigger samples produce less estimation errors and deflections, and allow users to make inferences about small but important population segments.

4 Applied Tool

We use a Data mining tool called WEKA to analyze data. First, we proceed to develop a model that allows explain the behavior showed by the four Italian communities, and how affects their computer activities and therefore their likes and purchase intention on the web. Figure 2 and 3 shows WEKA usage to discover the existent relation among four parameters associated to Italianitá.

We found in both cases that the RAI scholarship holders outside Italy showed a higher "Italianitá" regarding native Italians.

This can be explained by the Pygmalion effect because they resist losing their ancestors customs, and purchase decision is highly influenced by this effect induced by their relatives.



Weka Explorer: Visualizing italianitab

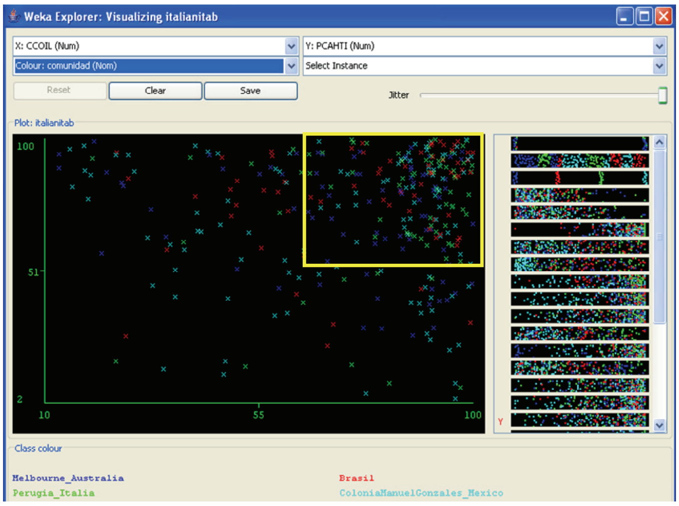
X: BMI (Num)

Y: CWI (Num)

Colour: comunidad (Nom)

Select Instance

Fig. 2. WEKA justifying the relation among the "some Italian" web site creation with the relation to download Italian music (superior cluster). Users that download music but do not have the intention to create a "some Italian" web site form another group (inferior cluster)



Weka Explorer: Visualizing italianitab

X: CCOIL (Num)

Y: PCAHTI (Num)

Colour: comunidad (Nom)

Select Instance

Fig. 3. Shows the relation to participate on a Chat on Italian with the purchase of items of Italian Origin

5 Results

We took in consideration RAI Internazionale scholarship holders of four Italian communities: Sample 1 (Melbourne, Australia), Sample 2 (Radamelli, Brasil), Sample 3

(Perugia, Italy) and Sample 4 (Manuel González Colony, Mexico City), to whose an instrument was applied by this organization, to identify different computer activities and purchase habits (See Table 1).

Table 1. Differences on computer skills by gender for the four Italian communities studied

	Sample 1		Sample 2		Sample 3		Sample 4	
	F	M	F	M	F	M	F	M
n	36	46	29	43	21	56	72	35
Online purchase of Italian books	22%	22%	21%	26%	29%	27%	14%	11%
Online purchase of any book	34%	31%	39%	21%	38%	37%	25%	26%
Having a PC	84%	87%	91%	80%	100%	96%	89%	91%
Creation of a "some Italian" web site	25%	50%	43%	63%	38%	66%	24%	29%
Write a Java Programm	39%	54%	4%	56%	29%	59%	7%	11%
Prepare a Power Point Presentation	78%	85%	75%	81%	95%	91%	66%	66%
Download documents on Italian with Acrobat	83%	93%	82%	98%	76%	93%	56%	66%
Sending photographs to relatives in Italy by means of e-mail	69%	91%	71%	79%	71%	87%	64%	66%
Install extra memory	31%	48%	29%	56%	48%	71%	21%	29%
Download Italian Music	72%	89%	89%	91%	81%	87%	80%	74%
Install an extra floppy drive	19%	57%	32%	40%	24%	66%	19%	20%
Send a static/silence greeting card	83%	67%	86%	74%	76%	70%	80%	60%
Send a animated/musical greeting card	81%	77%	89%	70%	76%	70%	79%	46%
Participate on Chats on Italian	61%	87%	75%	86%	71%	80%	73%	69%
Send an attachment by e-mail	91%	84%	100%	100%	100%	100%	87%	86%
Installation of a computers network	11%	17%	14%	44%	19%	60%	6%	6%
Purchase an italian origin item	80%	72%	71%	86%	71%	91%	76%	66%
Upgrade the PC's operating system	45%	47%	43%	72%	29%	79%	37%	34%
Research for online papers/assesments	88%	93%	100%	98%	90%	100%	98%	91%
Defragmentation of hard disk	48%	45%	75%	88%	52%	80%	47%	57%
Send a movie/video by e-mail	27%	33%	39%	65%	100%	62%	15%	31%
Purchase anything italian on "e-bay" (or other site)	35%	29%	18%	44%	33%	59%	28%	31%
Sell anything italian on "e-bay" (or other site)	19%	14%	11%	16%	19%	21%	10%	6%
Sum of PC Knowlege and Italianitá (mean)	5.6	7.2	12	14	11.1	14.9	9.7	9.4

The use of Data mining in social aspects has demonstrated being key part to corroborate the tendencies of a group with common ancestors (Pygmalion Effect), although on each group we identified factors that distorted the data analyzed in the answers (the factor of Lying is greater in women than in men), we found variations depending on the Italian community origin, see Table 2.

Table 2. Predictors to do computer activities and online purchase of books or items of Italian origin

Study 1	Women	Men
Skills for PC/Internet	0.43	0.15
Extraversion	0.23	0.10
Neuroticism	-0.30	0.19

Study 2	Women	Men
Skills for PC/Internet	0.44	0.26
Attitudes toward money:		
Power	0.14	0.04
Retention	0.06	0.33
Un confidence	-0.13	0.21
Anxiety	-0.25	0.26

Study 3	Women	Men
Skills for the PC/Internet	0.48	0.35
Psychopath	0.37	-0.15
Extroversion	-0.08	0.01
Neuroticism	0.21	-0.06
Factor of Lying	0.23	0.10

Study 4	Women	Men
Skills for PC/Internet	-0.02	0.11
Computers Anxiety	-0.07	0.21
Computers Attitude	0.07	0.21
Attitudes for the Internet	-0.03	0.04

6 Conclusions

There are an important number of questions that deserve additional research. One will be to find new information sources to mine about the users preferences. As we discuss earlier, researchers have investigated the hyperlinks structure, the electronic conversations and users' purchase records [12].

An area with great potential is the electronic usage of media, specifically, digital music. By analyzing what kind of music is someone listening, a system can deduce the songs, the singers and the genders the person prefers, and by using this information recommend additional songs and artists, to get the person in touch with people of similar interests. We made an approach on this direction with a system that allows users to view individual and group historical listening lists and define with this information new listening lists [1]. In [6] is shown a system that learns of the user preferences based on the music listened, after songs are selected to be play on a shared physical environment, based on the preferences of the whole people present. Meanwhile the user preferences are extracted from a large number of sources; the idea to combine different types of preferences starts to be important. In PHOAKS [19] preferences are extracted from web pages since USENET messages and then presented to the users. Showing how the users visualize this information. PHOAKS keeps the track on what pages the users did click (other type of implicit preference).

Development of general techniques to combine different types of preferences is now a challenge. Panzanni [15] presents a method to give weight to different types of contributions, however, if this is the best combination of methods and how to determine the proper weights is still a complex idea. Such system will combine the people advantages – applying the judgment to select the initial system of collections – and of computers to apply analysis of techniques to provide remarked information and to store updated collections. A similar tactic will be to use a search engine. Finally, this discussion shows that even a very large system, manually constructed from "base" pages can be improved perceptibly by providing additional characteristics, grouping pages on sites, and offering a user friendly interface.

7 Future Works

We are planning to apply a similar methodology to identify Mexican way of being, attitudes and purchasing habits over the Internet from Mexicans living abroad, specifically in the United States and in the European Union. They represent more than thirty million persons, almost a quarter of the total Mexican population, that represents the first international income for the Mexican Economy and a very interesting target market to explore for business opportunities.

By using a different instrument and samples from different places of the world, we are planning to compare two societies without sea and with a high migration level. Our basic question is: Can these societies develop similar behaviours?

Acknowledgements

We want to thank to Mendoza R & Rodríguez L. for sharing their Pygmalion Effect Model called "Talucicé" that allows explain beliefs on certain societies about the relation of the birds singing and yellow flowers with death relatives and about bio luminescent insects with divine messages like the Catholicism of Oriental Timor. To Ochoa P. for his economic support to purchase Social Data Mining books.

References

1. Amento B. Specifying Preferences based on User History. In Proceedings of CHI'2002, ACM Press. (2002)
2. Broedbeck K. The order of things: Activity-Centered Information Access. In Proceedings 7thICWWW'98. (1998)
3. Bush, V. As we may think. The Atlantic Monthly. (July 1945).
4. Card K. et al. The Information Visualizer, an Information Workspace for the World-Wide Web. CHI'96. (1996)

5. Daurov T. & Sebastianni M. Modelling Kazakh costumes using data mining. CA CCBR; Astana, Kazakhstán. (2005)
6. Fiore T. Visualization Components for persistent Conversations. In Proceedings of CHI'2001. (2001)
7. Han Jiawei, Implementing data mining for discover conduct patterns, Am-Psychol, Jan 32[1]: (2001) 57-66.
8. Hé Z. .& Milodragovich K. Discovering chinese descendents in Palé Island using Data Mining. CACCBR; Astana, Kazakhstán. (2005)
9. Logan S. Discovering induced social patterns using an Intelligent Dyoram for displayed.CACCBR; Kazakhstán. (2005)
10. Maes P. Social Information Filtering: Algorithms for Automating <<Word of Mouth>>. In Proceedings of CHI'95. (1995)
11. Munro J. & Höök K. Social Navigation of Information Space. Springer, (1999)
12. Nieto M.; Ochoa A. Applying dependences model to Data Mining software.CIIC-'02;Soto La Marina, México. (2002)
13. Oki, B; Momoi, Kaori & Zhang Ziyi. Collaborative Filtering to Weawe an Information Tapestry. Communications of the ACM, 35, (1992) 51-60.
14. Padméterakiris, A.; Gyllenhaal, J. & Ochoa A. Implementing of a Data Mining Algorithm for discovering Greek ancestors, using simetry patterns. Central Asia CCBR (Data Mining Workshop); Astana, Kazakhstán. (2005)
15. Pazzani M. & Diggory Cedric Learning Collaborative Information Filters. In Proceedings of ICML'98. (1998)
16. Pirolli, P. Life, Death and Lawfulness on the Electrical Frontier in Proceedings of CHI'97. (1997)
17. Rosenthal Robert Explain the Pygmalion effect in the school. Hamburg, Germany. (1971)
18. Tabrizi-Nouri H.; Tañón O.; Ianevski S. & Ochoa A. Explain mixtured couples support with Gini Coeficient. CACCBR (Data Mining Workshop); Astana, Kazakhstán. (2005)
19. Terveen L. Using Frequency-of-Mention in Public Conversations for Social Filtering. Proceedings CSCW'96. (1996)
20. Tochi K. & Amento B. Experiments in Social Data Mining: The TopicShow System In Proceedings CHI'03. (2003)
21. Toriello, A. & Hill W. Beyond Recommender Systems: Helping People Help Each Other. HCI in the new Millennium, Addison Wesley. (2001)
22. Viegas F. Chat circles. In Proceedings of CHI'99, ACM Press, (1999), 9-16.
23. Wexelblat, P. Footprints: History-Rich Tools for Information Foraging. In Proceedings of CHI'99. (1999)
24. Winograd T. An Information-Exploration Interface Supporting the Contextual Evolution of a User's Interests. In Proceedings of CHI'97 (1997)

Kernel Methods for Anomaly Detection and Noise Elimination

H. Jair Escalante

Instituto Nacional de Astrofísica Óptica y Electrónica
Luis Enrique Erro 1 Puebla, 72840, México
hugojair@ccc.inaoep.mx

Abstract. A kernel-based algorithm for *useful*-anomaly detection and noise elimination is introduced. The algorithm's objective is to improve data quality by correcting wrong observations while leaving intact the correct ones. The proposed algorithm is based on a process that we called "*Re-Measurement*" and it is oriented to datasets that might contain both kinds of rare objects: noise and useful-anomalies. Two versions of the algorithm are presented $R-V1$ and $R-V2$. Both algorithms generate new observations of a suspect object in order to discriminate between erroneous and correct observations. Noise is corrected while outliers are retained. Suspect data is detected by a kernel-based novelty detection algorithm. We presented experimental results of our algorithm, combined with KPCA, in the prediction of stellar population parameters a challenging astronomical domain, as well as in benchmark data.

1 Introduction

Real world data are never as perfect as we would like them to be and often can suffer from corruption that may affect data interpretations, data processing, classifiers and models generated from data as well as decisions made based on data. Affected data can be due to several factors including: ignorance and human errors, the inherent variability of the domain, rounding errors, transcription error, instrument malfunction, biases and, most important, rare but correct and useful behavior. For these reasons it is necessary to develop techniques that allow us to deal with affected data. As we can see corrupted data may be: noise (erroneous data) or anomalies (rare but correct data) and it would be very useful to differentiate between them from the rest of data. An expert can perform this process but it requires a lot of time investment which yields in expensive human-hour costs, from here arises the necessity of automate this task. However this is not an easy task since outliers and noise may look quite similar for an algorithm, then we need to add to such algorithm a more human-like reasoning. In this work the re-measurement idea is proposed, this approach consist of detecting "*suspect*" data and by generating new observations of these objects we can correct errors, while retaining anomalies for posterior analysis. This algorithm could be useful in several research areas, including: machine learning, data mining, pattern recognition, data cleansing, data warehousing, information retrieval and applications such as: security systems and medical diagnostic. In this work we oriented our efforts to improve data quality and prediction accuracy for machine learning problems, specifically for the estimation of stellar population parameters a domain in which the re-measurement algorithm is suitable to test.

Elimination of suspect objects have been widely used for most of anomaly detection methods [1–6], the popularity of this approach comes from the fact that they can alter calculated statistics, increase prediction error, turn more complex a model based on these data or possibly they introduce a bias in the process to which they are dedicated. However we should not eliminate an observation unless, like an expert, we can determine the incorrectness of the datum. This often is not possible for several reasons: human-hour cost, time investment, ignorance about the domain we are dealing and even uncertainty. Nevertheless if we could guarantee that an algorithm successfully will distinguish errors from rare objects with high confidence level the difficult task would be solved. Like an human does, an algorithm can confirm or discard a hypothesis by analyzing several samples of the same object.

Re-measurement is safer than elimination by several reasons: we can conserve rare objects and decide what to do about them, we can ensure that an anomaly is correct, we can eliminate the wrong objects from our dataset, we can be sure that a common instance will never be affected, all of these reasons make suitable the use of re-measurement instead of elimination in certain domains.

2 Estimation of Stellar Populations Parameters

In most of the scientific disciplines we are facing a massive data overload, astronomy is not the exception. With the development of new automated telescopes for sky surveys, terabytes of information are being generated. Recently machine learning researchers and astronomers have been collaborating towards the goal of automatizing astronomical analysis tasks. Almost all information about a star can be obtained from its spectrum, which is a plot of flux against wavelength. An analysis on galactic spectrum can reveal valuable information about star formation, as well as other physical parameters such as metal content, mass and shape.

Theoretical studies have shown that a galactic spectrum can be modeled with good accuracy as a linear combination of three spectra, corresponding to young, medium and old stellar populations, each with different metallicity and together with a model of the effects of interstellar dust in these individual spectra. Interstellar dust absorbs energy preferentially at short wavelengths, near the blue end of the visible spectrum, while its effects on longer wavelengths, near the red end of the spectrum, are small. This effect is called reddening in the astronomical literature. Let $f(\lambda)$ be the energy flux emitted by a star or group of stars at wavelength λ. The flux detected by a measuring device is then $d(\lambda) = f(\lambda)(1 - e^{-r\lambda})$, where r is a constant that defines the amount of reddening in the observed spectrum and depends on the size and density of the dust particles in the interstellar medium.

We also need to consider the redshift, which tells us how the light emitted by distant galaxies is shifted to longer wavelengths, when compared to the spectrum of closer galaxies. This is taken as evidence that the universe is expanding and that it started in a Big Bang. More distant objects generally exhibit larger redshifts; these more distant objects are also seen as they were further back in time, because the light has taken longer to reach us.

Therefore, a simulated galactic spectrum can be built given c_1, c_2, c_3, with $\sum_{i=1}^{3} c_i = 1, c_i > 0$ the relative contributions of young, medium and old stellar populations, respectively; their reddening parameters r_1, r_2, r_3, and the ages of the populations $a_1 \in \{10^6, 10^{6.3}, 10^{6.6}, 10^7, 10^{7.3}\}$ years, $a_2 \in \{10^{7.6}, 10^8, 10^{8.3} 10^{8.6}\}$ years, $a_3 \in \{10^9, 10^{10.2}\}$ years,

$$g(\lambda) = \sum_{i,m=1}^{3} c_i s(m_i, a_i, \lambda)(1 - e^{r_i \lambda})$$

with $m \in \{0.0004, 0.004, 0.008, 0.02, 0.05\}$ in solar units and $m_1 \geq m_2 \geq m_3$, finally we add an artificial redshift Z by:

$$\lambda = \lambda_0(Z + 1), 0 < Z \leq 1$$

Therefore, the learning task is to estimate the parameters: reddening (r_1, r_2, r_3), metallicities (m_1, m_2, m_3), ages (a_1, a_2, a_3), relative contributions (c_1, c_2, c_3), and redshift Z, from the spectra.

3 Methods

Kernel methods have demonstrated been useful tools for pattern recognition, dimensionality reduction, denoising, and image processing. In this work we used kernel methods for dimensionality reduction of spectral data. Also we used a kernel-based method for novelty detection. Furthermore the re-measurement algorithm differentiates anomalies from noise by using a kernel. In this section KPCA and the algorithm for anomaly detection used are briefly described.

3.1 Kernel PCA

Stellar populations data are formed with instances with dimensionality $d = 12134$, therefore, in order to perform experiments in feasible time we need a method for dimensionality reduction. Kernel principal component analysis (KPCA) [7] is a relative recent technique, which takes the classical PCA technique to the feature space, taking advantage of "kernel functions". This feature space is obtained by a mapping from the linear input space to a commonly nonlinear feature space F by $\Phi : \mathbf{R}^N \rightarrow F, x \mapsto X$.

In order to perform PCA in F, we assume that we are dealing with centered data, using the covariance matrix in F, $\overline{C} = \frac{1}{l} \sum_{j=1}^{l} \Phi(\mathbf{x}_j)\Phi(\mathbf{x}_j)^T$, we need to find $\lambda \geq 0$ and $\mathbf{v} \in F \setminus \{0\}$ satisfying $\lambda \mathbf{V} = \overline{C}\mathbf{V}$. After some mathematical manipulation and defining a $M \times M$ matrix K by

$$K_{i,j} := (\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)) \tag{1}$$

the problem reduces to $\lambda\alpha = K\alpha$, knowing that there exist coefficients $\alpha_i (i = 1, \ldots, l)$ such that $\lambda \mathbf{V} = \sum_{i=1}^{l} \lambda_i \Phi(\mathbf{x}_i)$.

Depending on the dimensionality of the dataset, matrix K in (1) could be very expensive to compute, however, a much more efficient way to compute dot products of the form $(\Phi(\mathbf{x}), \Phi(\mathbf{y}))$ is by using kernel representations $k(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}))$, which allow us to compute the value of the dot product in F without having to carry out the expensive mapping Φ.

Not all dot product functions can be used, only those that satisfy Mercer's theorem [8]. In this work we used a polynomial kernel (Eq. 2).

$$k(\mathbf{x}, \mathbf{y}) = ((\mathbf{x} \cdot \mathbf{y}) + 1)^d \tag{2}$$

3.2 Kernel Based Novelty Detection

In order to develop an accurate nose-aware algorithm we need first a precise method for novelty detection. We decided to use a novelty detection algorithm that has outperformed others in an experimental comparison [9]. This algorithm presented in [10] computes the center of mass for a dataset in feature space by using a kernel matrix K, then a threshold t is fixed by considering an estimation error (Eq. 3) of the empirical center of mass, as well as distances between objects and such center of mass in a dataset.

$$t = \sqrt{\frac{2 * \phi}{n}} * \left(\sqrt{2} + \sqrt{\ln \frac{1}{\delta}}\right) \tag{3}$$

where $\phi = \max(diag(K))$, and K is the kernel matrix of the dataset with size $n \times n$; δ is a confidence parameter for the detection process. This is an efficient and very precise method; for this work we used a polynomial kernel function (Eq. 2) of degree 1.

4 Re-Measurement Algorithm

Before introducing the re-measurement algorithm, the concept of the "re-measurement" process should be clarified. Given a set of instances: $X = \{x_1, x_2, \ldots, x_n\}$, with $x_i \in \mathbf{R}^n$ generated from a known and controlled process by means of measurement instruments or human recording. We have a subset $S \subset X$ of instances x_i^s with $S = \{x_1^s, x_2^s, \ldots, x_m^s\}$ and $m << n$ that according to a method for anomaly detection every $x_i^s, i = \{1, 2, \ldots, m\}$, is suspect to be a incorrect observation. Then, the re-measuring process consists of generating another observation $x_i^{s'}$ for each of the m objects, in the same conditions and using the same configuration that when the original observations were made.

The idea of re-measurement is based on the natural way in which a human clarifies his doubts; when a person is doubtful about the correctness of a datum he/shee can check the datum's validity by analyzing several observations of the same object. For example, in case our observations were pictures for face recognition, the re-measuring process would consists of taking another picture for every suspect object in our data set.

The re-measurement algorithm uses a confidence level value (*cl*) which tell us how rare a suspect object is. *cl* can indicates the number of re-measurements to perform for

Table 1. The $R - V1$ algorithm

```
Generate a dataset T which columns are attributes and rows instances
not-outlier-th:=0.99; outlier-th:=0.8;

0. Obtain the principal components ($PC_T$) for T
1. Identify t-suspect observations from $PC_T$

2. For each $i \in t$

- $cl_i := \ln(d_i * C)$
- $measurements_i := cl_i$-new observations of object $i$
- $k_{avg} := \frac{1}{cl} \sum_j^{cl} k(i, y_j)$,

- if ($k_{avg} \geq$not-outlier-th and $cl = 1$): return(not-outlier)
- else if($k_{avg} \geq$outlier-th):return(outlier)
- else return(noise)

3. For each $i \in t$ labeled as noise : $PC_T i := measurements_{irand}$
```

each suspect instance. cl value is obtained from the distance of the suspect objects to the center of mass in the feature space and it is defined in (4),

$$cl_i = \begin{cases} 1 & \text{if} \log(d_i * C) \leq 0 \\ \text{round}(\log(d_i * C)) & \text{otherwise} \end{cases} \tag{4}$$

Where d_i is the distance in feature space of the suspect instance x_1^s to the center of mass of the full data set, and C is a scaling constant.

For the anomaly-noise discrimination we decided to use a kernel, since kernels can be used to calculate similarity between objects [8]. Several kernels were tested but the kernel that best distinguished between instances was the extended radial basis function (5) with $\sigma = 0.25$. This kernel returns a 1 value if the instances (x, y) are identical or a value between $(0, 1)$, if they are different, that indicates how similar objects (x, y) are, near to 1 indicates more similitude.

$$k(x, y) = \exp - \left(\frac{\sqrt{\|x - y\|^2}}{2\sigma^2} \right) \tag{5}$$

Using this property of the kernel we generated simple rules to differentiate between noise, anomalies and common instances.

$$O = \begin{cases} not - outlier & \text{if } k_{avg} \geq 0.99 \text{ and } cl = 1 \\ outlier & \text{if } k_{avg} \geq 0.8 \\ noise & \text{otherwise} \end{cases}$$

where $k_{avg} = \frac{1}{cl} \sum_{j=1}^{cl} k(x, y_j)$, is the average of the kernel evaluations given a suspect instance x and its cl new measurements $y_1, \ldots, y_{cl}$ as inputs. In Table 1 the re-measurement algorithm applied to the prediction of stellar population parameters is presented, in step 0 we used KPCA as a preprocessing step. Next we applied the KB-novelty detection algorithm and with a little modification we forced the algorithm to return the top t-farther objects with their distance to the center of mass. cl-value is calculated and cl-new observations are generated and stored in $measurements_i$, then we calculate k_{avg}. This k_{avg} is compared with our thresholds and the algorithm decide the type of object, finally the erroneous objects are substituted by a random sample in $measurements_i$.

4.1 Reducing the Number of Re-Measurements

The proposed algorithm performs well but it requires of nearby 5 new samples to identify anomalies and above 2 for noisy objects, in some domains (including astronomy) the generation of a new instances is expensive and obtaining 5 or 4 new measurements is complicated. A little modification in the algorithm will overcome this difficulty, by a slightly change in our rules and by verifying them each time that a new measurement is generated we would need only a new sample to identify common instances and anomalies and at most 2 more to detect noise, we will call this algorithm $R - V2$, the new rules are:

$$O = \begin{cases} not - outlier & \text{if } d \geq 0.99 \text{ and } cl = 1 \\ outlier & \text{if } d \geq 0.8 \text{ and } cl \geq 2 \\ noise & \text{otherwise} \end{cases}$$

In Figure 1 the modification to the algorithm is shown. This time cl is used to complement the basic conditionals. Anomalies and common instances will be detected with only a new sample by using cl, while noise will be re-sampled a few times to discard confusions, finally all noise is substituted by a random sample.

5 Experimental Results

In order to test the performance of the re-measurement algorithms some experiments were performed. The stellar populations domain was used in the following way: in each experiment a dataset of 200 spectra is generated, 5% of this data is affected with additive(normal distributed) extreme noise (2.5% with positive mean and 2.5% with a negative one), another 5% of the data is shifted by a factor $(f \in R : 1 < f < 10)$ simulating useful-anomalies.

We compared accuracy using the mean absolute error (M.A.E.) obtained by a classifier builded with locally weighted linear regression (LWLR)[11]. LWLR belongs to the family of instance-based learning algorithms, these algorithms build query specific local models, which attempt to fit the training examples only in a region around a query point. For this work we considered a neighborhood of 80 points to approximate the target function.

In Table 2 percentage reduction error is presented for algorithms $R-V1$ and $R-V2$, we reported the average of 5 runs using a 10-fold cross validation.

There is an important reduction of error when we used KPCA, the maximum error reduction is attained when all of the suspect objects are eliminated, although we are loosing useful information too. Our algorithms reach accuracy only 2% down the

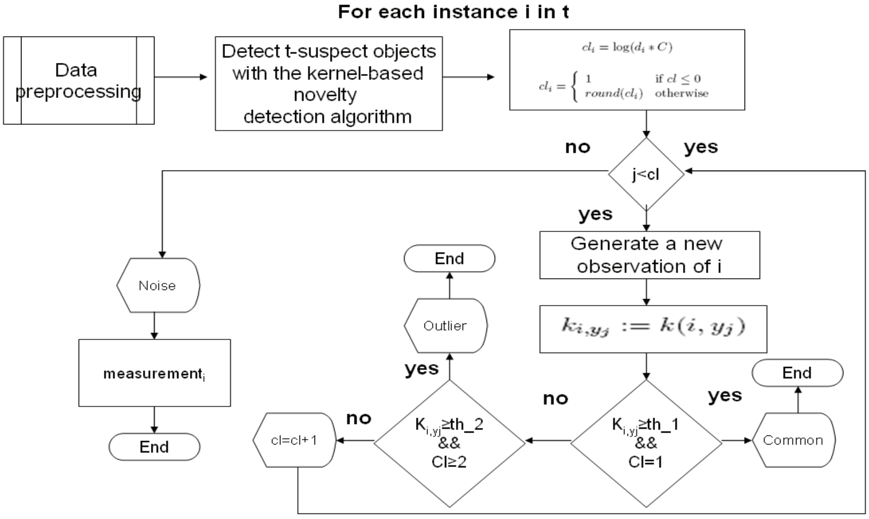


Fig. 1. Block diagram of the $R - V2$ algorithm

Table 2. Reduction percentage of M.A.E. for the prediction of stellar populations parameters regarding as baseline the M.A.E. obtained when the full-affected dataset was used, compared with using 10-KPCA, 10-KPCA when all suspect data is eliminated (KPCA-E) and using 10-KPCA with the re-measurement algorithm (KPCA-R), clean data was used

Method	Reddening	Metal	Ages	Contributions	Redshift	Average
			R-V1			
KPCA	17.29%	10.17%	13.58%	20.21%	-2.39%	11.29%
KPCA-E	20.88%	16.40%	19.19%	33.23%	29.90%	19.76%
KPCA-R	19.82%	14.30%	16.68%	27.14%	13.68%	16.13%
			R-V2			
KPCA	13.97%	1.57%	14.00%	20.60%	-4.30%	7.56%
KPCA-E	20.85%	7.29%	17.69%	31.85%	29.44%	14.96%
KPCA-R	16.98%	6.69%	15.53%	25.18%	16.24%	12.34%

Table 3. Reduction percentage of M.A.E. for the prediction of stellar population parameters, noisy data was used

Method	Reddening	Metal	Ages	Contributions	Redshift	Average
			R-V1			
KPCA	-1.71%	-1.25%	3.09%	-1.24%	-12.01%	-0.47%
KPCA-E	7.00%	5.41%	6.42%	13.02%	23.23%	7.88%
KPCA-R	2.24%	3.13%	7.61%	7.76%	5.29%	5.41%
			R-V2			
KPCA	2.66%	-0.62%	3.37%	3.07%	7.87%	1.84%
KPCA-E	9.80%	4.00%	10.80%	18.32%	25.40%	9.29%
KPCA-R	7.88%	3.57%	8.82%	11.46%	16.48%	7.16%

best performer, without eliminating any anomaly while correcting noisy objects. In real world domains however, data may be affected with low-level noise due to systematic errors, therefore, we performed experiments adding low-level noise to the full set of spectra and affected with extreme noise and anomalies as in the last experiment. In Table 3 we report results of this experiment, we observe the same behavior than in Table 2 however the results are diminished even, for the $R - V1$ algorithm, the KPCA result is worse than using the full dataset, it is possible that the number of PC's used is not the optimal for these affected data.

Accuracy improvement is significant when we used the re-measurement algorithm, however if we want to analyze data quality, accuracy may not be the best measure to compare. In Table 4 performance of the re-measurement algorithms $R-V1$ and $R-V2$ is shown.

As we see both algorithms detected and corrected 100% of the noise and none instance was confused. The anomaly detection rate was high although no perfect. There are not neither false anomalies nor false noisy objects detected. CLC is the cl value for common instances detected as suspicious and its value is obviously 1. CLO is the cl value for anomalies and it is of almost 5 for the $R-V1$ algorithm and of 1 for $R-V2$, this means that only a new sample was needed for identify anomalies and common instances while for the case of noisy objects cl value (CLN) is of 1.5. This results on the cl values confirm that the selection of cl (4) is adequate. Processing time decreases about 25% for the $R - V2$ algorithm in this artificial dataset which yields in saving some seconds, although for real data the decrement could be of hours.

Last three rows on Table 4 show the performance of the KB-algorithm for novelty detection. We present the F-measure value obtained by such algorithm. This measure is based on recall $\mathbf{R} = \frac{TP}{(TP+FN)}$ and precision $\mathbf{P} = \frac{TP}{(TP+FP)}$ and it is defined as $F = \frac{2*R*P}{(R+P)}$, where TP is for true positives, TN is for true negatives, FP is for false positives and FN is for false negatives. $F-$measure express with a real number in [0,1] the performance of an outlier detection method. We forced the novelty detection algorithm to return the top 30 points farther the center of mass and this is a reason of because $F-$measure value is not perfect, however a look on the TP and FP rates is more useful.

Besides the good performance of the re-measurement algorithms in the astronomical domain, we had doubts about the performance of our algorithms in other domains.

Table 4. Performance of the re-measurement algorithms: $R - V1$ and $R - V2$

Algorithm		**R-V1**		**R-V2**
Parameter / Data	**Clean**	**Noisy**	**Clean**	**Noisy**
Re-Measurement				
Anomalies Detected	90%	100%	80%	86.6%
Noise Detected	100%	100%	100%	100%
Confused	0	0	0	0
False Anomalies	0.33	1.33	0	0
False Noise	0	0	0	0
CLC-value	1	1	1	1
CLO-value	4.86	4.26	1	1
CLN-value	1.5	1.37	1.6	1.5
Time(s)	77.57	87.64	55.34	57.32
Novelty Detection				
TP	19.33	20	19	20
FP	0.67	0	1	0
F-measure	0.77	0.8	0.76	0.8

For this reason we performed experiments on ten data sets from the UCI repository[12], the datasets used are briefly described in Table 5.

In this experiments we used only the $R - V2$ algorithm since it is the best performer on the above experimentation, moreover we performed experiments with noise only, since it allow us to simulate the re-measurement process. Each data set was normalized to the range $[0, 1]$ and it was affected as with the astronomical domain. Results on accuracy for these datasets are show in Table 8, while the performance results are presented in tables 6 and 7.

As we can see the $R - V2$ algorithm performance on UCI data is similar to the observed in the astronomical data. There is an accuracy improvement in all of the datasets

Table 5. UCI Datasets description

ID	**Name**	$\#_{Cases} - \#_{Features}$	**Output**	**# Affected**
W	Wine	$178 - 13$	3-Discrete	18
G	Glass	$214 - 9$	Real	21
H	Boston Housing	$506 - 13$	Real	51
A	Auto	$32 - 7$	Real	3
I	Iris	$150 - 4$	3-Discrete	15
M	Machine CPU	$209 - 6$	Real	21
L	Lymphography	$148 - 18$	4-Discrete	15
C	Breast Cancer	$683 - 9$	2-Discrete	68
B	Bio Med	$194 - 5$	2-Discrete	19
Ab	Abalone	$1000 - 8$	Real	100

Table 6. Performance of the $R - V2$ algorithm for the UCI datasets. We present: CLC, CLO and CLN as before, outliers detected (O.D.), noise detected (N.D.), confusions(Conf.), false outliers (F.O.) and false noise (F.N.)

Dataset	W	G	H	A	I	M	L	C	B	Ab
CLN	3.33	3.66	3.58	3.67	4.29	4.15	2.92	3.82	4.21	3.88
O.D.(%)	100	100	96.15	100	100	100	100	100	100	100
N.D.(%)	100	100	98.67	100	100	100	100	100	96.67	98
F.O.	0	0	0	0	0	3	0	0	2	11

Table 7. Performance of the kernel-based novelty detection algorithm used. We present the number of suspect observations detected, true positives and false negatives, recall, precision and $F-$measure

Dataset	W	G	H	A	I	M	L	C	B	Ab
Suspect	27	32	76	5	23	31	22	102	29	150
TP	18	21	49.66	3	15	21	15	68	18.66	99
FN	0	0	1.33	0	0	0	0	0	0.33	1
Rec	1	1	0.97	1	1	1	1	1	0.98	0.99
Prec	0.76	0.66	0.65	0.6	0.65	0.68	0.68	0.67	0.64	0.66
F	0.8	0.79	0.78	0.75	0.79	0.81	0.81	0.8	0.78	0.79

Table 8. Reduction percentage of M.A.E. for each dataset, when suspect data is eliminated(E) and when we used the $R - V2$ algorithm, compared with the prediction of LWLR using the full data. In last 3 rows, novelty detection algorithm performance is presented

Dataset	W	G	H	A	I	M	L	C	B	Ab
	Red %									
E	0.54	15.76	15.77	24.61	5.23	13.89	6.29	1.83	0.85	5.86
$R - V2$	10.86	14.74	34.54	28.37	4	28.62	5.63	1.75	2.24	5.08

when we used $R - V2$ even in some results our algorithm improved the elimination of suspect data. The algorithm needed only a new sample to identify outliers and common instances and nearby 4 to detect noise. There are not confusions and the false outliers rate was low, although we had false outliers only in two datasets. Performance of the kernel-based algorithm for novelty detection again is almost perfect.

6 Conclusions

We have introduced the re-measurement process as an option for useful-anomaly and noise differentiation. Two kernel based algorithms were presented, $R-V2$ needs only a new sample to identify anomalies and at most two for noisy objects. Anomalies remain unaffected while noise is substituted in an almost automated process (an user may be needed to generate the new measurements). Our algorithms are model data independent and can be generalized for non real valued domains, since they are based on kernels.

Experimental results on an astrophysics domain as well as on benchmark data are presented, our algorithm combined with KPCA improves prediction accuracy and data quality for the astronomical domain, while for the UCI data the same pattern is observed showing the generalization ability of our algorithm. This algorithm could be useful in domains requiring of highly-reliable data or in those in which the novelty is more interesting than the rest of the objects.

Acknowledgments This work was partially supported by CONACYT under grant 181498.

References

1. Carla Brodley. Identifying mislabeled training data. *JAIR*, 11:131–167, 1999.
2. R. T. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In *20th ICVLDB*, pages 144–155, 1994.
3. D. Gamberger, N. Lavrač, and C. Grošelj. Experiments with noise filtering in a medical domain. In *Proc. 16th ICML*, pages 143–151. Morgan Kaufmann, San Francisco, CA, 1999.
4. David Tax and Robert Duin. Data domain description using support vectors. In *Proceedings of the European Symposium on Artificial Neural Networks*, pages 251–256, 1999. ISBN 2-600049-9-X.
5. B. Schölkopf, o Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution, 1999.
6. George H. John. Robust decision trees: Removing outliers from databases. In *Proc. of the 1st ICKDDM*, pages 174–179, 1995.
7. B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. In *Neural Computation 10*, pages 1299–1319, 1998.
8. R. Herbrich. *Learning Kernel Classifiers*. MIT press, first edition, 2002. ISBN 0-262-08306-X.
9. H. Jair Escalante. Resampling algorithms for machine learning. Master's thesis, Instituto Nacional de Astrofísica Óptica y Electrónica, to appear, 2005.
10. J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
11. Christopher Atkeson, Andrew Moore, and Stefan Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1-5):11–73, 1997.
12. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

Noise-Aware Algorithms for Analysis of Galactic Spectra

H. Jair Escalante[1] and Olac Fuentes[2]

[1] Instituto Nacional de Astrofísica Óptica y Electrónica
Luis Enrique Erro 1 Puebla, 72840, México
hugojair@ccc.inaoep.mx
[2] University of Texas at El Paso
500 West University Avenue, El Paso TX, USA
ofuentes@utep.edu

Abstract. We introduce a novel learning algorithm for noise elimination. Our algorithm is based on the re-measurement idea for the correction of erroneous observations and is able to discriminate between noisy and noiseless observations by using kernel methods. We apply our noise-aware algorithms to the prediction of stellar population parameters, a challenging astronomical problem. Experimental results adding noise and useful anomalies to the data show that our algorithm provides a significant reduction in error, without having to eliminate any observation from the original dataset.

1 Introduction

Real world data are never as good as we would like them to be and often can suffer from corruption that may affect data interpretation, data processing, classifiers and models generated from data as well as decisions based on them. On the other hand, data can also contain useful anomalies, which often result in interesting findings, motivating further investigation. Thus, unusual data can be due to several factors including: ignorance and human mistakes, the inherent variability of the domain, rounding and transcription errors, instrument malfunction, biases and, most important, rare but correct and useful behavior. For these reasons it is necessary to develop techniques that allow us to deal with unusual data.

Data cleaning is a well studied task in many areas dealing with databases, nevertheless, this task requires a large time investment. Indeed, between 30% to 80% of the data analysis task is spent on cleaning and understanding the data [1]. An expert can clean the data, but this requires a large time investment, growing with the number of observations in the data set, which results in expensive costs. From here arises the need to automate this task. However, this is not easy, since useful anomalies and noise may look quite similar to an algorithm. For this reason we need to endow to such algorithm with more human-like reasoning. In this work the re-measurement idea is proposed; this approach consist of detecting suspect data and, by analyzing new observations of these objects, substitute errors while retaining anomalies and correct data for a posterior analysis. This idea is based on the natural way in which a human clarifies his/her doubts when he/she is not sure about the correctness of a datum. When a person suspects of

© A. Gelbukh, S. Torres, I. López (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 19, 2006, pp. 81-92

an object's observation, a new observation or many more can be obtained to confirm or discard the observer's hypothesis.

The proposed methods could be useful in areas such as machine learning, data mining, pattern recognition, data cleansing, data warehousing and information retrieval. In this work we oriented our efforts to improve data quality and prediction accuracy for machine learning problems, specifically, for the estimation of stellar population parameters, an interesting domain in which an algorithm based on re-measuring is suitable to test.

The paper is organized as follows: in the next section we present a brief survey of related works. In Section 3 we introduce the astronomical domain used in this work; in Section 4 the kernel methods that we used are described. In Section 5 the proposed algorithms are introduced. In Section 6 experimental results evaluating the performance of our algorithms are presented. Finally, we summarize our findings and discuss future directions for this work in Section 7.

2 Related Work

Recent approaches for data cleansing do not distinguish between useful anomalies and noise, they just eliminate the detected suspect data [2–8]. However, we should not eliminate a datum unless we can determine that it is invalid. This often is not possible for several reasons, including: human-hour cost, time investment, ignorance about the domain we are dealing with and even inherent uncertainty. Nevertheless, if we could guarantee that an algorithm will successfully distinguish errors from correct observations, the difficult problem would be solved. As a human does, an algorithm can confirm or discard a hypothesis by analyzing several measurements of the same object.

The idea of requesting new observations as a strategy for data cleansing has been little explored. Here we present some related works that deal with anomaly detection and data cleaning.

In [9] an interactive method for data cleaning that uses the optimal margin classifier (OMC) is presented. The OMC is used to identify suspect data, suspect observations are shown to an expert in the domain, who then decides their validity.

Prototype [10] and instance selection [11] implicitly can eliminate instances degrading the performance of instance-based learning algorithms. Other algorithms saturate a dataset with the risk of eliminating all objects that could define a concept or class, these methods include the use of instance pruning trees [8] and the saturation filtering algorithm [4]. Ensembles of classifiers had been successfully used to identify mislabeled instances in classification problems [12, 5, 13], however, once again the identified instances are deleted from the data set.

In the outlier/anomaly detection area there are many published works, however, these approaches are intended only for the detection of rare data. The anomaly detection problem has been approached using statistical [14] and probabilistic knowledge [15], distance and similarity-dissimilarity functions [16–18], metrics and kernels [19], accuracy when dealing with labeled data, association rules, properties of patterns and other specific domain features.

Variants and modifications to the support vector machine algorithm have been proposed, trying to isolate the outlier class: in [20] an algorithm to find the support of a dataset, which can be used to find outliers, is presented; in [6] the sphere with minimal radius enclosing most of the data is found and in [7] the correct class is separated from the origin and from the outlier class for a given data set.

There are many more methods for anomaly detection than the presented here, however, we have only presented some of the representative ones. What is important to notice is that at the moment there are automated approaches for data cleaning that are concerned with the elimination of useful data.

3 Estimation of Stellar Populations Parameters

In most of the scientific disciplines we are facing a massive data overload, and astronomy is not the exception. With the development of new automated telescopes for sky surveys, terabytes of information are being generated. Such amounts of information need to be analyzed in order to provide knowledge and insight that can improve our understanding about the evolution of the universe. Such analysis becomes impossible using traditional techniques, thus automated tools should be developed. Recently, machine learning researchers and astronomers have been collaborating towards the goal of automating astronomical data analysis tasks. Such collaborations have resulted in the automation of several astronomical tasks. These works include galaxy classification [21], prediction of stellar atmospheric parameters [22] and estimation of stellar population parameters [23].

In this work we applied our algorithms for the prediction of stellar population parameters: ages, relative contribution, metal content, reddening and redshift. In the remaining of this section the data used are briefly described.

3.1 Analysis of Galactic Spectra

Almost all the relevant information about a star can be obtained from its spectrum, which is a plot of flux against wavelength. An analysis of a galactic spectrum can reveal valuable information about stellar formation, as well as other physical parameters such as metal content, mass and shape. The accurate knowledge of these parameters is very important for cosmological studies and for the understanding of galaxy formation and evolution. Template fitting has been used to carry out estimates of the distribution of age and metallicity from spectral data. Although this technique achieves good results, it is very expensive in terms of computing time and therefore can be applied only to small samples.

Modeling Galactic Spectra Theoretical studies have shown that a galactic spectrum can be modeled with good accuracy as a linear combination of three spectra, corresponding to young, medium and old stellar populations, see Figure 1, with their respective metallicity, together with a model of the effects of interstellar dust in these individual spectra. Interstellar dust absorbs energy preferentially at short wavelengths, near the blue end of the visible spectrum, while its effects on longer wavelengths, near the red end of the spectrum, are small. This effect is called reddening in the astronomical literature. Let $f(\lambda)$ be the energy flux emitted by a star or group of stars at wavelength λ. The flux detected by a measuring device can be approximated as $d(\lambda) = f(\lambda)(1 - e^{-r\lambda})$, where r is a constant that defines the amount of reddening in the observed spectrum and depends on the size and density of the dust particles in the interstellar medium.

We also need to consider the redshift, which tells us how the light emitted by distant galaxies is shifted to longer wavelengths, when compared to the spectrum of closer galaxies. This is taken as evidence that the universe is expanding and that it started in a Big Bang. More distant objects generally exhibit larger redshifts; these more distant objects are also seen as they were further back in time, because the light has taken longer to reach us.

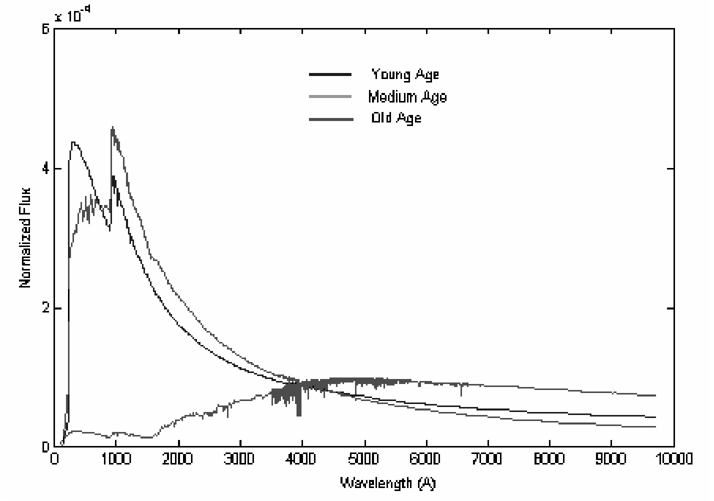


Fig. 1. Stellar spectra of young, intermediate and old populations.

We build a simulated galactic spectrum given constants c_1, c_2, and c_3, with $\sum_{i=1}^{3} c_i = 1$ and $c_i > 0$, that represent the relative contributions of young, medium and old stellar populations, respectively; their reddening parameters r_1, r_2, r_3, and the ages of the populations $a_1 \in \{10^6, 10^{6.3}, 10^{6.6}, 10^7, 10^{7.3}\}$ years, $a_2 \in \{10^{7.6}, 10^8, 10^{8.3} 10^{8.6}\}$ years, and $a_3 \in \{10^9, 10^{10.2}\}$ years,

$$g(\lambda) = \sum_{i,m=1}^{3} c_i s(m_i, a_i, \lambda)(1 - e^{r_i \lambda})$$

with $m \in \{0.0004, 0.004, 0.008, 0.02, 0.05\}$ in solar units and $m_1 \geq m_2 \geq m_3$, finally we add an artificial redshift Z by:

$$\lambda = \lambda_0(Z + 1), 0 < Z \leq 1$$

Therefore, the learning task is to estimate the parameters: reddening (r_1, r_2, r_3), metallicities (m_1, m_2, m_3), ages (a_1, a_2, a_3), relative contributions (c_1, c_2, c_3), and redshift Z, from the spectra.

4 Kernel Methods

Kernel methods have been shown to be useful tools for pattern recognition, dimensionality reduction, denoising, and image processing. In this work we use kernel methods for dimensionality reduction, novelty detection and anomaly-noise differentiation.

4.1 Kernel PCA

Stellar populations data are formed with instances with dimensionality $d = 12134$, therefore, in order to perform experiments in feasible time we need a method for dimensionality reduction. Kernel principal component analysis (KPCA) [24] is a relative recent technique, which takes the classical PCA technique to the feature space, taking advantage of ”kernel functions”. This feature space is obtained by a mapping from the linear input space to a commonly nonlinear feature space F by $\Phi : \mathbf{R}^N \rightarrow F, x \mapsto X$.

In order to perform PCA in F, we assume that we are dealing with centered data, using the covariance matrix in F, $\overline{C} = \frac{1}{l}\sum_{j=1}^{l} \Phi(\mathbf{x}_j)\Phi(\mathbf{x}_j)^T$, we need to find $\lambda \geq 0$ and $\mathbf{v} \in F \setminus \{0\}$ satisfying $\lambda\mathbf{V} = \overline{C}\mathbf{V}$. After some mathematical manipulation and defining a $M \times M$ matrix K by

$$K_{i,j} := (\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)) \tag{1}$$

the problem reduces to $\lambda\alpha = K\alpha$, knowing that there exist coefficients $\alpha_i (i = 1, \ldots, l)$ such that $\lambda\mathbf{V} = \sum_{i=1}^{l} \lambda_i \Phi(\mathbf{x}_i)$.

Depending on the dimensionality of the dataset, matrix K in (1) could be very expensive to compute, however, a much more efficient way to compute dot products of the form $(\Phi(\mathbf{x}), \Phi(\mathbf{y}))$ is by using kernel representations $k(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}))$, which allow us to compute the value of the dot product in F without having to carry out the expensive mapping Φ.

Not all dot product functions can be used, only those that satisfy Mercer’s theorem [25]. In this work we used a polynomial kernel (Eq. 2).

$$k(\mathbf{x}, \mathbf{y}) = ((\mathbf{x} \cdot \mathbf{y}) + 1)^d \tag{2}$$

4.2 Kernel based novelty detection

In order to develop an accurate nose-aware algorithm we need first a precise method for novelty detection. We decided to use a novelty detection algorithm that has outperformed others in an experimental comparison [26]. This algorithm presented in [19] computes the center of mass for a dataset in feature space by using a kernel matrix K, then a threshold t is fixed by considering an estimation error (Eq. 3) of the empirical center of mass, as well as distances between objects and such center of mass in a dataset.

$$t = \sqrt{\frac{2 * \phi}{n}} * \left(\sqrt{2} + \sqrt{\ln \frac{1}{\delta}}\right) \tag{3}$$

where $\phi = \max(diag(K))$, and K is the kernel matrix of the dataset with size $n \times n$; δ is a confidence parameter for the detection process. This is an efficient and very precise method; for this work we used a polynomial kernel function (Eq. 2) of degree 1.

5 Noise-Aware Algorithms

Before introducing the noise-aware algorithms, the *re-measuring* process must be clarified. Given a set of instances: $X = \{x_1, x_2, \ldots, x_n\}$, with $x_i \in \mathbf{R}^n$ (generated from a known and controlled process by means of measurement instruments or human recording), we have a subset $S \subset X$ of instances x_i^s with $S = \{x_1^s, x_2^s, \ldots, x_m^s\}$ and $m << n$ that, according to a method for anomaly detection are suspect to be incorrect observations. Then, the re-measuring process consists of generating another observation $x_i^{s'}$ for each of the m objects, in the same conditions and using the same configuration that when the original observations were made.

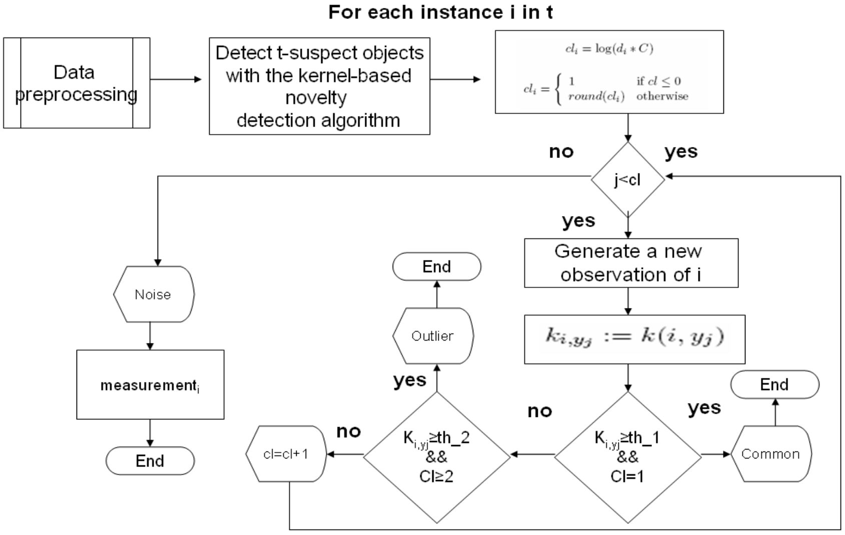


Fig. 2. Block diagram of the base noise-aware algorithm.

In Figure 2 the base noise-aware algorithm is shown. The data preprocessing module includes dimensionality reduction, scaling data, feature selection or similar necessary processes. In the next step, suspect data are identified by using an anomaly detection method. Then, a confidence level cl is calculated; this cl indicates how rare an object is, and it can be used to determine the number of new measurements to obtain for each of the suspicious instances. cl is obtained from the distance of the suspect instances to the center of mass of the full data set, and it is defined in Eq. (4).

$$cl_i = \begin{cases} 1 & \text{if } \log(d_i * C) \leq 0 \\ \text{round}(\log(d_i * C)) & \text{otherwise} \end{cases} \tag{4}$$

Where d_i is the distance in feature space of the suspect instance x_1^s to the center of mass of the full data set, and C is a scaling constant.

For the anomaly-noise discrimination we decided to use a kernel, since kernels can be used to calculate similarity between objects [25]. Several kernels were tested, but the kernel that best distinguished among dissimilar instances was the extended radial basis function (Eq. 5) with $\sigma = 0.25$.

$$k(x, y) = \exp\left(\frac{-\sqrt{\|x - y\|^2}}{2\sigma^2}\right) \tag{5}$$

We generated simple rules to discriminate among noise, outliers and common instances. If an object is correct, the algorithm leaves that object intact, otherwise, the noisy observation is substituted by one in the new measurements. The generated decision rules were:

$$O = \begin{cases} not - outlier & \text{if } k_{avg} \geq 0.99 \text{ and } cl = 1 \\ outlier & \text{if } k_{avg} \geq 0.8 \text{ and } cl \geq 2 \\ noise & \text{otherwise} \end{cases}$$

where $k_{avg} = \frac{1}{cl} \sum_{j=1}^{cl} k(x, y_j)$, is the average of the kernel evaluations given a suspect instance x and its cl new measurements $y_1, \ldots, y_{cl}$ as inputs. As we can see, outliers and common instances will be detected with only a new observation, while noise will be re-measured a few times, finally all of the noise is substituted by a correct measurement or in other approach by the average of the re-measurements.

The algorithm from Figure 2 can be used for cleaning datasets, eliminating all of the noise and retaining correct observations. Now we have to describe how to take advantage of it to improve the results of a machine learning task.

In Figure 3, the base noise-aware algorithm is adapted to predict the stellar population parameters in the astronomical data, using locally-weighted linear regression LWLR [27], a well known machine learning algorithm.

We have divided the data cleaning process into two phases: training and testing. Data cleaning in training is just what we have descibed before in the base algorithm. Data cleaning for testing data is somewhat different, in this setting we have a new data set of p (unseen) new observations. Then, the algorithm uses the distance of each test observation to the center of mass of the improved training set to determine the set of suspicious test data. Suspect observations are re-measured. Then, the erroneous observations are differentiated from correct observations and wrong data are substituted by the average of their measurements, while for correct rare observations the original measurement was used. In the case of correct observations we could also use the average of the measurements, which, as we will see, results in better accuracy in experiments with noisy data.

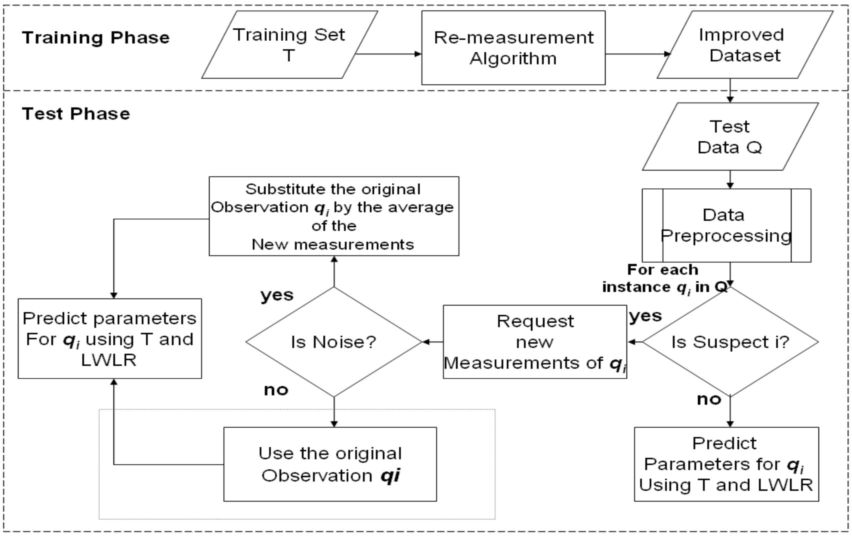

Training Phase
Training Set T
Re-measurement Algorithm
Improved Dataset
Test Phase
Test Data Q
Data Preprocessing
For each instance q_i in Q
Is Suspect i?
yes
no
Request new Measurements of q_i
Is Noise?
yes
no
Substitute the original Observation q_i by the average of the New measurements
Predict parameters For q_i using T and LWLR
Use the original Observation qi
Predict Parameters for q_i Using T and LWLR

Fig. 3. Block diagram of a noise-aware machine learning algorithm.

6 Experimental Results

We performed several experiments in order to test the performance of our algorithms. In each experiment we generated a dataset of 200 observations for training and 3 datasets of 100 observations for testing. We used LWLR as learning algorithm considering a neighborhood of 80 objects. All results presented here are averages over the three test datasets.

In the first experiment we tested the base noise-aware algorithm inserting noise and outliers to the datasets. For this experiment all of the data were affected with low-level noise; 5% of the data were affected with extreme gaussian noise with zero mean, and varying the value of σ^2, as shown in Figure 4. Furthermore, 5% of the data were affected by inserting useful anomalies.

Useful anomalies were simulated in a realistic way. Commonly, redshift values lie in the range $(0 \leq Z \leq 0.4)$; redshifts higher than 1 are useful anomalies for astronomers. In astronomy, locating galaxies with redshifts over 2 is very useful for galaxy evolution research. We simulated in 5% of the data redshifts between 2 and 4 $(2 \leq Z \leq 4)$.

The experiment consists of applying the algorithm from Figure 3, to the prediction of the stellar population parameters, using a training dataset previously improved with the algorithm from Figure 2. Results of these experiments are shown in Table 1; the mean absolute error (M.A.E.) reduction is presented. We report results using different configurations for training and testing. We can see that the best results are those obtained when the training set has been improved with our algorithm. The best result was obtained when the original (affected) test data were used, however, there is not a sig-

Table 1. Percentage of M.A.E. reduction for the different configurations on the training and test sets. Noisy is the original (affected) data set, and noise-aware is the data that have been previously improved with our algorithm. The first column indicates the training data used, while the first row indicates the test data used.

Training/Test	*Noisy*	*Noise-Aware*
Noisy	0	0.01
Noise-Aware	4.19	3.46

nificant difference. What is important to notice is that an improvement in the training set results in an improvement of the prediction accuracy in the test sets. Something remarkable, that is not shown in the tables, is that the noise-aware algorithm detected 14 of the 15 total artificially-added anomalies on the test datasets. Furthermore, 100% of the noisy observations were corrected, which would result in data quality improvement without a loss of useful information. In order to determine how much the heuristics implemented in the noise-aware algorithms help to improve the accuracy, we performed another experiment. In the following experiments we compared the performance of our algorithm with one that re-measures randomly, without repetition; again, we divided the data into training and test sets. For these experiments, all of the data sets were affected with 4 different noise levels (gaussian, with mean zero and varying the standard deviations), see Figure 4. The experiment consists of comparing the noise-aware algorithm form Figure 3 with one that randomly chooses instances to re-measure. In this scenario, we have the capability of performing R new measurements. Therefore, the random method performs a new measurement of R objects chosen randomly, without repetition. On the other hand, the noise-aware algorithm (Figure 3) iterates on the data

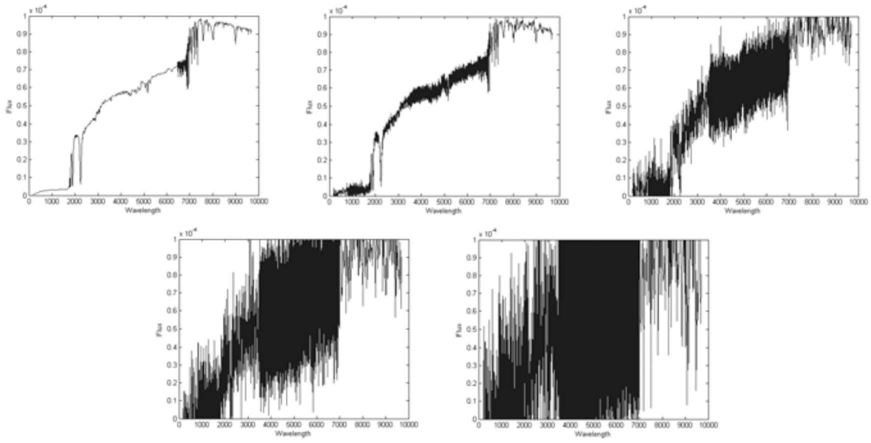


Fig. 4. Sample spectra with the different levels of noise added. In all of the figures, the noise is Gaussian with zero mean and varying the standard deviation in each case.

set, until R re-measurements are made. That is, in each iteration the algorithm identifies, re-measures and corrects erroneous observations. We substituted the noisy observations by the average of the new measurements, due to the nature of the noise added. The results for the training phase, with $R = 200, 100, 66$, are presented in Table 3. We can

Table 2. Percentage of M.A.E. reduction, Noisy is the original (affected) dataset; noise-aware is the dataset that has been improved with our algorithm; random is the dataset improved with the method that re-measures randomly.

	$R = 200$	
	%	Time
Noisy	0	0
Random	-6.35	273.86
Noise-Aware	15.54	298.56
	$R = 100$	
Noisy	0	0
Random	-7.11	138.9
Noise-Aware	14.82	154.38
	$R = 66$	
Noisy	0	0
Random	-1.39	90.79
Noise-Aware	9.65	147.40

Table 3. Percentage of M.A.E. reduction in the training phase for different values of R, for the random method and the noise-aware algorithm.

Training/Test	*Noisy*	*Random*	*Noise-Aware*
Noisy	0.00	2.88	2.12
Random	-3.4	-5.86	-2.07
Noise-Aware	6.15	7.01	6.61

see from Table 3 that there is a clear improvement by using our algorithm instead of the one that re-measures randomly. Indeed, when the random method is used there is a slight decrease in accuracy. The improvement is large when we iterate our algorithm until 200 new measurements are made. Moreover, the difference in processing time is small. The performance of the algorithms in the test sets can be seen in Table 2. Again, we presented different configurations for training and testing. From Table 2, we can observe that the best result was obtained when we used the improved training data. For testing, the best result was obtained when the random algorithm was used. However, the difference in accuracy is small. We performed the same experiment but instead of using the original measurement for low and medium noise affected observations, we used the average of the new-measurements. Results of this experiment are shown in Table 4. We can see that there is a clear improvement in our algorithm when all of the suspect data

Table 4. Percentage of M.A.E. reduction for the different configurations of training and test sets. In this experiment all of the suspect observations were substituted by the average of the new measurements in the noise-aware algorithm.

Training/Test	*Noisy*	*Random*	*Noise-Aware*
Noisy	0.00	0.21	2.81
Random	-2.46	-2.7	-1.18
Noise-Aware	5.69	6.74	10.88

were substituted. With this modification, the best result is obtained when both training and testing data were improved with our algorithm. The improvement is around 11% in accuracy. The behavior of the random method was similar to that in Table 2.

7 Conclusions and Future Work

We have presented the re-measuring idea as a method for the correction of erroneous observations in corrupted datasets without eliminating potentially useful observations. Experimental results showed that the use of a noise-aware algorithm in training sets improves prediction accuracy using LWLR as learning algorithm. The algorithms were able to detect and correct 100% of the erroneous observations and around 90% of the artificial outliers, which resulted in a data quality improvement. Furthermore, we have shown that the noise-aware algorithms outperformed a method that re-measures randomly in the prediction of stellar population parameters, a difficult astronomical data analysis problems.

Present and future work includes testing our algorithms on benchmark datasets to determine their scope of applicability. Also, we plan to apply noise-aware algorithms in other astronomical domains as well as in other areas, including bioinformatics, medical diagnosis, and image analysis.

Acknowledgments This work was partially supported by CONACYT under grant 181498.

References

1. Tamraparni Dasu and Theodore Johnson. *Exploratory Data Mining and Data Cleaning*. Probability and Statistics. Wiley, 2003.
2. Carla Brodley. Identifying mislabeled training data. In *Journal of Artificial Intelligence Research*, volume 11, pages 131–167, 1999.
3. Raymond T. Ng and Jiawei Han. Efficient and effective clustering methods for spatial data mining. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 144–155. Morgan Kaufmann Publishers, 1994.
4. Dragan Gamberger, Nada Lavrač, and Ciril Grošelj. Experiments with noise filtering in a medical domain. In *Proceedings of the 16th International Conference on Machine Learning*, pages 143–151. Morgan Kaufmann, San Francisco, CA, 1999.
5. Sofie Verbaeten and Anneleen Van Assche. Ensemble methods for noise elimination in classification problems. In *Multiple Classifier Systems*, volume 2709 of *Lecture Notes in Computer Science*, pages 317–325. Springer, 2003.

6. D. Tax and R. Duin. Data domain description using support vectors. In *Proceedings of the European Symposium on Artificial Neural Networks*, pages 251–256, 1999. ISBN 2-600049-9-X.
7. B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution. In *Technical Report 99-87, Microsoft Research*, 1999.
8. George H. John. Robust decision trees: Removing outliers from databases. In *Proceedings of the 1st. Int. Conf. on KDDM*, pages 174–179, 1995.
9. N. Matic I. Guyon and V. Vapnik. Discovering informative patterns ans data cleaning. In *Advances in Knowledge Discovery and Data Mining*, pages 181–203, 1996.
10. David B. Skalak. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *ICML*, pages 293–301, 1994.
11. Henry Brighton and Chris Mellish. Advances in instance selection for instance-based learning algorithms. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining*, pages 153–172, 2003.
12. Carla E. Brodley and Mark A. Friedl. Identifying and eliminating mislabeled training instances. In *AAAI/IAAI, Vol. 1*, pages 799–805, 1996.
13. David Clark. Using consensus ensembles to identify suspect data. In *KES*, pages 483–490, 2004.
14. Vic Barnett and Toby Lewis. *Outliers in Statistical Data*. John Wiley and Sons, 1978. ISBN 0-471-99599-1.
15. J. Kubica and A. Moore. Probabilistic noise identification and data cleaning. In *Technical Report CMU-RI-TR-02-26, CMU*, 2002.
16. Andreas Arning, Rakesh Agrawal, and Prabhakar Raghavan. A linear method for deviation detection in large databases. In *Knowledge Discovery and Data Mining*, pages 164–169, 1996.
17. Edwin M. Knorr and Raymond T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24th International Conference in Very Large Data Bases, VLDB*, pages 392–403, 1998.
18. Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 427–438, Dallas, Texas, USA, 2000. ACM. ISBN 1-58113-218-2.
19. John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
20. B. Schölkopft, A. Smola, R. Williamson, and P. Bartlett. New support vector algorithms. In *Neural Computation*, volume 12, pages 1083 – 1121, 2000.
21. Jorge de la Calleja and Olac Fuentes. Machine learning and image analysis for morphological galaxy classification. *Monthly Notices of the Royal Astronomical Society*, 349:87–93, 2004.
22. Olac Fuentes and Ravi K. Gulati. Prediction of stellar atmospheric parameters using neural networks and instance-based learning. In *Experimental Astronomy 12:1*, pages 21–31, 2001.
23. Olac Fuentes, Thamar Solorio, Roberto Terlevich, and Elena Terlevich. Analysis of galactic spectra using active instance-based learning and domain knowledge. In *Proceedings of IX IBERAMIA, Puebla, Mexico*. Lecture Notes in Artificial Intelligence 3315, 2004.
24. B. Schölkopf, A. Smola, and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. In *Neural Computation*, volume 10, pages 1299–1319, 1998.
25. Ralf Herbrich. *Learning Kernel Classifiers*. MIT press, first edition, 2002. ISBN 0-262-08306-X.
26. H. Jair Escalante. Noise-aware machine learning algorithms. Master's thesis, Instituto Nacional de Astrofísica Óptica y Electrónica, January 2006.
27. Christopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1-5):11–73, 1997.

On the Compression of Geography Markup Language

Nieves R. Brisaboa*, Antonio Fariña*, Miguel Luaces*, José R. Rios Viqueira† and José R. Paramá*

*Database Lab., Univ. da Coruña,
Facultade de Informática, Campus de Elviña s/n,
15071 A Coruña, Spain.
† Dept. Electronics and Computer Science
Univ. Santiago de Compostela, Fac. de Fisica, Campus Universitario Sur,
15782 Santiago de Compostela, Spain.
{brisaboa, fari, luaces, parama}@udc.es and joserios@usc.es

Abstract. The Geography Markup Language (GML) is a standard XML-based language that enables the representation and easy interchange of geographic data between Geographic Information Systems (GIS).
In this paper, we explored the compressibility of GML performing some empirical experiments over real GML corpus using a wide set of well-known compressors. In particular, the main characteristics of GML are first described. Next, it is shown how these characteristics can be exploited to achieve a better compression rate on GML files. We use these ideas to design a specific parser and a strategy to compress the representation of geographic objects (their coordinates in a map). Finally, to check the correctness of our hypothesis, the same GML files are compressed by applying the new parser and coordinates encoding strategy.

1 Introduction

In the last years, the technology underlying Spatial Databases and Geographic Information Systems (GIS) has undergone a great development. As a consequence, public administrations and governments are increasingly demanding tools and applications based on Geographic Information Systems (GIS) technology. This technology uses spatial extensions for object-relational databases to store the geometry of geographical objects. That is, these applications not only use standard alphanumeric data, but also geographic data to represent both the shape and the situation in the territory of different geographic objects (plots, roads, buildings, rivers, etc.).

The use of many different commercial off-the-shelf GIS tools, supporting different geographic data formats, makes the interchange of data among different systems difficult. To overcome these inter-operability problems, a standardization effort has been undertaken by the Open Geospatial Consortium (OGC). An

[⋆] Partially supported by MCyT (PGE and FEDER) grant(TIC2003-06593) and Xunta de Galicia grant(PGIDIT05SIN10502PR).

© A. Gelbukh, S. Torres, I. López (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 19, 2006, pp. 93-104

important result of such an effort is the definition of the Geography Markup Language (GML), which is an XML language that enables the representation of objects with both alphanumeric and geographic attributes of any kind.

The last version of this language has recently become a draft of the ISO technical committee 211, which is in charge of standardization in the field of digital geographic information.

Geographic datasets usually contain many geographic objects, each of them defined in terms of a long list of floating point coordinates. Therefore, representing such datasets with GML usually leads to huge text files and consequently to serious efficiency problems in their storage and transmission over the Internet.

To the best of our knowledge, no compression techniques have ever been used to compress GML. Probably this is due to the fact that the systematic use of GML is new, in fact the ISO committee is still standardizing GML. On the other hand, a good part of the GIS community does not have a computer science background.

In this paper, the application of various compression techniques to GML files has been investigated. In particular, in Section 2 the main characteristics GIS and GML are first described. Next, in Section 3, it is shown how these characteristics can be exploited to achieve a better compression rate on GML files and, also in this section, we present strategies to parser and to process coordinates in order to obtain a better compression by taking advantage of those features. Finally, in Section 4, the compressibility of GML is tested using real GML corpus and a wide set of well known compressors. We first use all the compressors directly over the GML files to have a baseline to compare with, and then, we explore the utility of our strategies to improve the compression ratio.

There are several well-known classic compression techniques such as Huffman [18] or Ziv-Lempel [27]. However, the widespread of the web caused the development of a wide range of new compression techniques designed to save storage space and/or transmission time.

Some of these compression methods [22, 21, 12, 14, 13] are *statistical* (also known as "zero-order substitution" methods). Statistical compression techniques split the original text into *symbols* and each symbol is represented (in the compressed text) by a unique *codeword*. Compression is achieved by assigning shorter codewords to more frequent symbols. These techniques need to compute the frequency of each original symbol and then a coding scheme is used to assign a codeword to each symbol.

Other compression methods [17, 27] are based on the use of a *dictionary*. These methods substitute the occurrence of strings in the text by pointers (all of them with the same length) to the correct entry in the dictionary. The longer the strings, the better the compression. These methods take advantage of the *co-occurrence* of characters or words because it permits, in general, longer strings and shorter dictionaries.

Both kinds of compression methods can be either static (vocabulary/dictionary fixed in advance), semistatic or dynamic. Semistatic statistical compression methods are also known as two pass methods [22, 21, 12, 14] since they perform two passes over the original text. In the first pass, they compute the frequency of each symbol, then the coding schema of each method assigns a codeword to each source symbol in the vocabulary and finally, in the second pass, each input symbol in the original text is substituted by its corresponding codeword. Something similar happens with semistatic dictionary-based methods [17].

Classic statistical compression methods used characters as input symbols. However, Moffat in [20] proposed the use of words instead of characters as the symbols to be compressed. By using words instead of characters compression ratios improve drastically, because the distribution of words is much more biased than that of characters.

There are still other well-known compression methods that follow different strategies. For example, arithmetic compressors [16] represent the whole text with a single number depending on the probability of the words in the text. PPM [11] is a statistical compressor that uses arithmetic encoding. Finally, since XML has been proposed as a standard to represent documents some research has been oriented to explode the structure of the documents to get a better compression. SCM-PPM [10] is an example of this kind of compressors.

2 Geographic Information Systems: Basic Concepts

For the purposes of the present paper, *geographic or spatial data* is any kind of data with a reference to *Geographic Space*, i.e., to the Earth surface. Two main categories of *spatial data* have been identified in the GIS literature [23, 24], *spatial objects* and *spatial fields*. A *spatial object* is described by a collection of properties of conventional data types (integer, real, string, etc.) and a collection of properties of spatial data types (point, line, surface), the latter defining its position and shape in *Geographic Space*. Examples of spatial objects are cities, rivers, roads etc.

A *spatial field* is a mapping from the positions of a subset of *Geographic Space* to the domain of some conventional property. A *spatial field* may be either discrete or continuous. In a *discrete field*, a single conventional value is assigned to each different element of a finite collection of either points or lines or surfaces. Examples of discrete spatial fields are the soil type, vegetation type, etc. In the case of *continuous fields* each point of an infinite subsect of *Geographic Space* is mapped to a value of a conventional domain. Examples of continuous spatial fields are the temperature, elevation above see level, etc.

A Spatial Data Infrastructure (SDI) includes a collection of spatial data sources and services. The services are useful to discover, access and process the data in the data sources integrated in the SDI. To guarantee the interoperability between the services of a SDI, the definition of standards was something mandatory. The Open Geospatial Consortium (OGC) [9] has proposed standard specifications for the interfaces of web services that enable a uniform access to heterogeneous collections of spatial data sources. In particular, a Web Feature Service (WFS) may be used to access repositories of *spatial objects* (*Features with geometry* in OGC terminology) in the web. Similarly, a Web Coverage Service (WCS) may be used to access repositories of *spatial fields* (*coverages* in OGC terminology). Finally, a Web Map Service (WMS) may also be used to generate maps (images in formats such as JPG, PNG, SVG, etc.) by assigning visualization styles (colors, line widths, etc.) to the data retrieved from WFSs and WCSs. Commercial and open source tools already exist that implement the standard WFS [1, 4, 8, 6, 7], WCS [1, 3] and WMS [1, 8, 6, 7] interfaces above. To represent the data retrieved by the WFS, the OGC has also defined an XML language called Geography Markup Language (GML). Versions 1.0, 2.0 and 2.1 of GML support the representation of both conventional and spatial properties of *spatial objects*. Among other pieces of functionality, support for the representation of *spatial fields* has been added to GML in versions 3.0 and 3.1. Therefore, GML can now also be used to represent data retrieved by a WCS. Version 3.1 of GML is currently a draft of the ISO technical committee 211, which is in charge of the standardization in the area of digital geographic information. Finally, it is remarked that most of the GML currently used is coded with version 2.1, therefore the remainder of this paper is restricted to this version. According to this, the following subsection gives a brief description of the representation of spatial objects in GML 2.1.

It is interesting to point out that all these standards are in full use since its definition because they were fully accepted by the GIS community. In Europe, the INSPIRE (INfrastructure for SPatial InfoRmation in Europe) [5] initiative of the European Commission intends to trigger the creation of an European SDI, which integrates national, regional and local SDIs of the member states. To establish a legal framework for the creation and operation of such and SDI, a directive of the European Parliament and of the Council has also been proposed by the Commission. Such a directive will force public administrations at national, regional and local level to follow the INSPIRE principles making mandatory the use of the described standards including GML. Therefore in the near future more and more GML files will be exchanged among spatial databases, at less in Europe. As a consequence, the interest of developing techniques for GML compression is clear.

2.1 Representing Spatial Objects in GML

To model *Geographic space*, a *Coordinate Reference System* (CRS) [19] is required, which assigns a tuple of numeric coordinates to each of its positions. Various types of CRSs have been used since the first cartographic representations of the Earth surface. As an example, in a *geographic CRS* positions are expressed in terms of latitude and longitude coordinates. Notice however that because of the curvature of the Earth surface, in order to display its positions in a 2-D flat surface (screen, paper sheet, etc.), a projection is needed. Thus, in a *projected CRS*, positions are expressed in terms of 2-D cartesian coordinates, and a projection is used to map each such position to the relevant position in *Geographic Space*. The *Universal Transverse Mercator* (UTM) is a well-known example of such a projection. The generation of UTM coordinates for each position in the Earth surface is next informally illustrated. First, the shape of the

Earth is approximated by an ellipsoid and each of its positions projected to a horizontal cylinder that surrounds the spheroid. Then, the cylinder is unrolled from north to south and the resulting flat surface partitioned into 60 vertical zones, each of them of 6 degrees wide. The equator splits each such zone into two subzones, north and south. A different origin of coordinates is assigned to each of the above subzones. Both north and south subzones have their origin X coordinate 500,000 meters west from the central meridian of the zone. However, the origin Y coordinate of north and south subzones is placed, respectively, at the equator and 10,000,000 meters south from the equator. These origins guarantee that both X and Y coordinates of positions are always positive distances.

In order to represent in a computer the infinite number of points contained in a spatial object, discrete finite representations had to be developed. Two main types of such discrete representations can be found in the literature [23, 24], namely *vector* and *raster representations*. Roughly speaking, in a *vector representation*, the coordinates of the CRS are approximated by either integer or floating point numbers. A *point* is represented by a pair of (x, y) coordinates of the CRS. *Lines* are approximated by sequences of line segments, each of them represented by its two end-points. A *surface* is represented by the vector approximation of its boundary, which is in the general case composed of an exterior boundary line and the boundary of a collection of holes. In a *Raster representation*, *Geographic Space* is partitioned into a collection of disjoint surfaces, called cells, of the same size and shape (usually squared). A *spatial object* is approximated by a set of such cells. Finally, it is well-known that a *vector representation* achieves a more precise representation of *spatial objects* and *discrete spatial fields* with a low storage cost, whereas, *raster representations* are better suited for the representation of *continuous fields* [24]. Therefore, the representation used by GML for spatial objects is vector-based.

To illustrate the use of GML in a realistic example consider the schema shown in Figure 1. It enables the representation of collections of municipalities in GML.

```
<?xml version="1.0"?>
<xsd:schema targetNamespace="http://www.core06.mx/Ex"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:ex="http://www.core06.mx/Ex"
    xmlns:gml="http://www.opengis.net/gml" elementFormDefault="qualified">
 <import namespace="http://www.opengis.net/gml" schemaLocation="feature.xsd"/>
 <import namespace="http://www.w3.org/1999/xlink" schemaLocation="xlink.xsd"/>
 <element  name="Example"  type="AbstractFeatureCollectionType"
    substitutionGroup="gml:_FeatureCollection"/>
 <element  name="Municipality"  type="ex:MunicipalityType"
    substitutionGroup="gml:_Feature"/>
 <complexType name="MunicipalityType"> <complexContent>
   <extension base="gml:AbstractFeatureType"><sequence>
      <element name="id" type="long"/> <element name="name" type="string"/>
      <element name="population" type="float"/>
      <element name="geo" type="gml:PolygonPropertyType"/>
   </sequence> </extension>
 </complexContent> </complexType>
</schema>
```

Fig. 1. XML Schema of a GML 2.1 Document.

The *import* element in line (6) specifies the location of the file “feature.xsd” that contains the schema for the definition of GML features (it is reminded that a feature with geometry is the term used by the OGC to denote a spatial object). Next, two elements are defined in the example schema. First an “Example” element is declared as a subtype of the GML type *AbstractFeatureCollectionType*, which enables the representation of collections of spatial objects. It is remarked here that the OGC defines that a collection of features is also a feature. Next, a “Municipality” element is declared. The definition of its type “MunicipalityType” follows in lines (12-18). It is noticed that “MunicipalityType” is defined as a subtype of the GML *AbstractFeatureType*, which is also defined in “feature.xsd” and enables the representation of spatial objects. Besides the general purpose properties already defined in type *AbstractFeatureType*, “MunicipalityType” contains also application specific conventional properties of each municipality. These are the “id” and “name” of the municipality. Finally, a spatial property “geo” of spatial data type polygon is also included in the “MunicipalityType”.

```
( 1)<?xml version="1.0"?>
( 2)<Example xmlns="http://www.opengis.net/gml"
( 3)         xmlns:ex="http://www.core06.mx/Ex">
( 4)<boundedBy><Box><coord><X>308787.49</X><Y>4744080.86</Y></coord>
( 5)  <coord><X>315101.36</X><Y>4748098.29</Y></coord></Box></boundedBy>
( 6)<featureMember> <ex:Municipality>
( 7)  <ex:id>1</ex:id><ex:name>Mazaricos</ex:name>
( 8)  <ex:geo><Polygon srsName="EPSG:23031"><outerBoundaryIs>
( 9)   <LinearRing><coordinates>
(10)      309440.29,4744357.47 309038.81,4744668.04 308787.49,4745676.36
(11)      310118.86,4746562.93 310363.24,4747445.11 311109.89,4747560.09
(12)      312741.84,4748098.29 313455.53,4747914.19 309440.29,4744357.47
(13)   </coordinates></LinearRing></outerBoundaryIs></Polygon></ex:geo>
(14)</ex:Municipality> </featureMember>
(15)<featureMember> <ex:Municipality>
(16)  <ex:id>2</ex:id><ex:name>Oroso</ex:name>
(17)  <ex:geo><Polygon srsName="EPSG:4230"><outerBoundaryIs>
(18)   <LinearRing><coordinates>
(19)      -99.0249938,56.6880477 -99.0258258,56.687203 -99.0255803,56.6863047
(20)      -99.0248452,56.6854873 -99.0241594,56.685209 -99.023588,56.6851012
(21)      -99.0223964,56.6851823 -99.018169,56.6861353 -99.0249938,56.6880477
(22)   </coordinates></LinearRing></outerBoundaryIs></Polygon></ex:geo>
(23) </ex:Municipality> </featureMember>
(24)</Example>
```

Fig. 2. Example of a GML 2.1 Document.

Based on the GML schema described above, a GML document containing a collection of two municipalities is shown in Figure 2. First, the minimum bounding rectangle that contains all the spatial objects in the represented collection is declared in the *BoundedBy* element in lines (4-5). It is noticed that the coordinates of such a rectangle are given in the “EPSG:23029” CRS, name given by the European Petroleum Survey Group to the UTM Zone 31 north. The *BoundedBy* element is mandatory for feature collections. Next the first member municipality of the collection is represented. Line (7) represent the conventional properties of the municipality ("id" and "name"). Next, lines (8-13) define the spatial property "geo", whose data type is polygon and whose coordinates are again coded with the UTM CRS. The second municipality of the collection is represented in lines (15-23). Now, the coordinates of the polygon of this municipality are coded in degrees of latitude and longitude.

In the GML example in Figure 2, it can be observed that an important part of the document is occupied by numeric coordinates. Obviously, the number of coordinates depends much on the precision of the represented data. Usually, the percentage of document occupied by coordinates is low in collections of point spatial objects and large in collections of lines and surfaces, reaching in some cases an amount of more than 80% of the file.

3 Compressing GML

GML documents are a special kind of textual documents, therefore we though that compressing them as any other textual document, without considering their specificity, would lead to losing compression ratio. Our hypothesis is that there are some GML features that can be exploited to increase its compressibility. In the next section we show the empirical data that proves such affirmation. In this section, we describe the GML characteristics that we have exploited to improve the compression ratio. Those features are:

1. GML files include many tags and numbers, which represent coordinates. On the other hand, the alphanumeric part represents data extracted from the columns of the non-spatial part of the database. Therefore it can not be considered as a natural language document. Our hypothesis was that the word frequency distribution in GML documents would be very different of those typical in natural language documents.
   We checked this hypothesis and computed the word frequency distribution of GML text and we found that it could be approximated by the Zipf distribution [26], but the Θ parameter of Zipf distribution, that has a value between 1.2 and 1.6 in natural language text, has in GML text the average value of 0.6. Therefore, we thought that the usual word-based compressors used in natural language documents such as [20–22, 14, 12] would not be efficient, as we will prove later.
2. In natural language documents, words can be usually identified by a sequence of alphabetic characters ('a'..'z', 'A'..'Z', '0'..'9') and everything between words is considered a separator. However, GML is cluttered with tags, and each appearance of a specific alphanumeric attribute of the database is marked with the tag corresponding to the name of such attribute. That is, a tag is written before and after the value itself. Tags are always written between '<' and '>'. Furthermore, other characters such as '=' or '_' appear systematically after some words.

From this characteristic, we argued that parsers used in word-based compressors are not adequate, and we designed a new parser taking into account the specific use in GML of tags and other usual symbols. Then we empirically checked the new parser as shown in Section 4.

3. GML is an XML-based language, as a consequence it can have many indentations, produced by long strings of blank spaces. Such spaces are useless, except to make the documents easier to read by humans. In the compression research field, spaces are always respected as any other character in the text, and they can not be removed. However, in GML, it does not make sense to keep all these spaces. In fact, some WFS, such as "Deegree" [1] do not insert spaces in the GML files, while other applications like JUMP [25], introduce a lot of them. Notice that spaces are not useful to split words because information in GML appears in the middle of the appropriate tags that always can be identified because they start and end with '<' and '>' respectively.
   Hence, we decided to study how the spaces (removed or not) affect the different compressors tested, but our hypothesis was that it would be useful to remove the spaces in the compressed text. We present these results in Section 4.
4. Coordinates represent a significative part (that can be even the 95%) of GML files. Such coordinates describe the points of the vectors that conform the geometry of each spatial object. Coordinates representing a spatial object can be a long sequence of numbers. However, some of the digits of these numbers will be equal in every coordinate, since coordinates represent points in a spatial object geometry, using UTM or Longitude or Latitude coordinates systems. Evidently, a point in an object may not be very far from others in the same object, and therefore, the most significative digits are usually the same since all points of a object belong to the same geographic area.
   On the other hand, coordinates rarely appear in two objects at the same time or twice in the same object. Therefore, introducing each coordinate value in the vocabulary as a new entry, and encoding it with a specific code would not produce compression. Word-based statistical compressor do exactly that and therefore we argue that this kind of compressors will obtain very poor compression ratios.
   To deal with this characteristic we decided to design a strategy to process the coordinates in order to improve the results of different compressors. Specifically, we focused our attention in the word-based statistical compressors since we thought that those compressors would be the most affected ones.

3.1 Strategies to Improve GML Compression

The strategies followed to improve GML compression were:

1. A lossless compressor must be able to compress and later decompress a text obtaining an exact representation of the original text, but we considered that in this case the spaces are meaningless and therefore we decided that removing them before the compression would be acceptable in GML. Empirical results show the gain obtained by doing that.

2. We built a new parser adapted to GML. This parser identifies the characters that are useful in GML to split words. Tags are identified as an unit, and repeated attributes ending with specific symbols such us "=" are identified as a single word. Section 4 presents the effect of this new parser.
3. To avoid the large amount of digits repeated in the coordinates, we decided to represent each coordinate as a difference to the previous one. In this way, the more significative numbers, representing the general geographic area where all the spatial points of the object are placed, are not repeated because after the first coordinate all the others are differences with respect to the previous one. This leads to save a big amount of space since we need less digits to represent differences than to represent full coordinates. On the other hand, to reduce the space used to represent numbers (inside coordinates), we decided to use a compact representation which uses only 4 bits for each number. This gives 12 codes that are enough to represent the 10 digits plus the symbols '$+'$ and '$-'$, needed to represent the sign of the difference with the previous coordinate and, at the same time, to identify the beginning of each coordinate.

Summarizing, we preprocess the GML text removing spaces before to start the compression with any compressor. Then, to improve the compression of word-based statistical compressor, we substituted coordinates by its differences and we represent numbers with a compact representation of 4 bits. In addition, to improve the compression of word-based statistical compressors, we designed a specific parser to identify the best words candidates. This parser identifies as a single word each whole set of compacted differences of consecutive coordinates describing the vectorial representation of a spatial object. Therefore this (maybe long) array of bytes, each byte representing two numbers of the differences among coordinates, is introduced as a single entry into the vocabulary and encoded with a byte oriented codeword (that usually has 3 bytes, or 4 if the file is huge). At the end, the vocabulary is compressed using character-based bit oriented Huffman.

4 Empirical Data

The main target of this section is to test the compressibility of GML files. We used some real GML files extracted from the EIEL Geographic Information System accesible in the web [2]. This system include a huge amount of information about the infrastructures of some Spanish municipalities. We extracted GML files form the following tables: Contour lines (CL), Plots (P), Municipalities borders (MB), Municipalities information(MI), Water supply network (WS), Roads (R) and Road Stretches (RS).

We started checking the amount of spaces that are included in the GML files using different applications. Using *Deegree* to generate the GML file about *Municipalities Borders* we obtained a file of 3,394,257 bytes, while *JUMP*, using the same table of the database, produces a GML file of 7,130,665 bytes. Then we compressed these two files obtaining the results shown in Table 1. Since those spaces are meaningless and produce a loss in compression, we decided to remove them (which is equivalent to use *Deegree* to obtain the GML files).

In Table 2, columns 1 and 2 show the name and size of the files without spaces. Column 3 represents the percentage of the file size occupied by coordinates. Notice that the space occupied by coordinates changes drastically depending on the complexity of the shape of the spatial objects included in the GML file.

All files were compressed with different compressors, some were general-purpose compressors, some were text compressors and finally, some were specially designed to compress XML files. As general-purpose compressors, we included the dictionary-based compressor *gzip* [27], one of the most frequently used compressors, *bzip2*, which is based in the Burrows-Wheeler Transform [15], and an *arithmetic* compressor [16] customized to use characters as symbols.

We also used two word-based and byte oriented statistical compressors: Plain Huffman (PH)[21], a Huffman-based compressor and End Tagged Dense Code (ETDC) [14], less efficient but faster and easier to implement, which is a compressor of the Dense family [12, 13].

Finally, we included *SCMPPM* [10] which is an adaption of the well known predictive compressor PPM [11] specifically adapted to compress XML files.

FILE	**size**	gzip	bzip2	scmppm	arith	ETDC	PH	Class. Huff.
JUMP	7,130,665	1,153,291	904,308	874,192	2,638,057	2,540,224	2,455,558	2,694,058
Deegree	3,413,795	996,145	890,498	810,741	1,749,144	2,381,037	2,229,043	1,802,206

Table 1. Compression removing and without removing spaces (sizes in bytes).

FILE	**size**	**coord.**			scm-		**unmodified**		**xml parser**		**Coord. Diff**	
	(kb.)	(%)	gzip	bzip2	ppm	arith	ETDC	PH	ETDC	**PH**	ETDC	PH
CL	4,322	67.90	20.34	15.83	15.35	60.52	38.88	37.68	33.45	33.44	15.38	15.37
MB	3,414	91.74	29.18	26.09	23.75	51.24	69.75	67.14	45.57	45.53	21.11	21.07
P	8,081	9.85	3.92	3.02	2.67	64.89	27.47	27.28	13.75	13.52	11.59	11.36
MI	12,770	95.98	31.73	29.40	27.69	48.83	53.72	52.63	46.46	46.45	23.53	23.52
WS	25,642	58.43	22.39	19.95	18.45	63.78	45.62	44.78	36.96	36.90	24.45	24.39
R	54,736	82.90	34.90	31.92	30.26	56.06	57.19	56.84	41.82	41.76	29.20	29.15
RS	81,332	55.03	24.05	21.22	19.94	65.16	48.16	47.92	35.12	35.05	26.81	26.75

Table 2. Compression of spaceless files using modified parser and coordinate processing.

First we compare all these techniques without any modification, just to compare the regular version of these compressors when they are applied to GML files. Table 2 shows the compression ratios achieved by the compression techniques included in our study (columns 4 to 9). General-purpose compressors gzip and bzip2 obtain good compression ratios, between 20% and 30%, except in the case of the *Plots* file, where the small percentage of the file occupied by coordinates produces a much better compression. It can be observed that text compressors have problems with GML files, because the streams of coordinates are not compressible with these compressors. Finally, the best results are obtained by the SCMPPM, although it does not take special care of the coordinates. Columns 10 to 13 show the results after adapting the parsers of the text compressors ETDC and PH to the GML characteristics. Furthermore, the influence of using the compact representation of the coordinates as differences can be seen by comparing the compression obtained when such preprocessing of the coordinates is done (columns 12 and 13) with the compression obtained when the compression of coordinates follows the standard procedure of the rest of the file (columns 10 and 11).

As it can be seen, when the coordinates are processed, ETDC and PH become closer to the compression ratios of SCMPPM. On the other hand, ETDC and PH are better than other alternatives when the percentage of space occupied by coordinates is significative.

5 Conclusions

The results presented in this paper demonstrate that we should devote attention to the characteristics of GML in order to compress it efficiently. The ideas shown here are only a first approximation to the problem, obviously they need to be improved, specially the efficiency of the compression/decompression process.

On the other hand, GML is automatically produced by software modules (possibly conforming with the WFS standard) and can be automatically read by other software modules (such as those following the WMS standard). We think that it could be convenient to develop applications including the WFS or WMS standards and compressors, to use a compressed version of GML by pipelining the compressor with the web service. That is, the output of a GML source could be compressed by the appropriate module, and before such a compressed file is provided as input to a GML consumer, a decompression module could decompress it to provide the information in plain form.

Another research line that should be undertaken is the possibility of searching patterns directly in the compressed text. This problem has been tackled, in natural language, by several researchers [21, 14, 12]. However, in natural text. GML represents spatial objects, so different applications could take advantage of the possibility of searching directly into the GML files. However, searching inside of GML involves new constraints and characteristics not present in usual text retrieval tasks.

We believe that this work opens a new field with new challenges to the compression research field. The compression of GML files has different constraints and possibilities that are not present in the compression of other kind of files such as text, DNA, images or music. Furthermore, presumably GML applications will attract more demand day after day and then, more different applications could benefit from the use of good compressors.

References

1. Deegree. URL: http://deegree.sourceforge.net/.
2. The EIEL project. URL: http://www.dicoruna.es/webeiel/.
3. ESRI arcGIS server. URL: http://www.esri.com/software/arcgis/arcgisserver/.
4. The GeoServer project. URL: http://geoserver.sourceforge.net/html/.

5. INSPIRE: INfrastructure for SPatial InfoRmation in Europe. URL: http://inspire.jrc.it/home.html.
6. Intergraph geomedia web map. URL: http://imgs.intergraph.com/gmwp/.
7. Mapinfo mapxtreme. URL: http://extranet.mapinfo.com/products/.
8. Mapserver. URL: http://mapserver.gis.umn.edu/.
9. OGC: Open geospatial consortium. URL: http://www.opengeospatial.org/.
10. J. Adiego, G. Navarro, and P. de la Fuente. Scm: Structural contexts model for improving compression in semistructured text databases. In *Proc. SPIRE 2003*, LNCS 2857, pages 153–167. Springer, 2003.
11. T. Bell, J. Cleary, and I. Witten. Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications*, 32(4):396–402, 1984.
12. Nieves R. Brisaboa, Antonio Fariña, Gonzalo Navarro, and Maria F. Esteller. (s,c)-dense coding: An optimized compression code for natural language text databases. In *Proc. SPIRE 2003*, LNCS 2857, pages 122–136, 2003.
13. Nieves R. Brisaboa, Antonio Fariña, Gonzalo Navarro, and José Paramá. Simple, fast, and efficient natural language adaptive compression. In *Proceedings SPIRE 2004*, LNCS 3246, pages 230–241. Springer, 2004.
14. Nieves R. Brisaboa, Eva L. Iglesias, Gonzalo Navarro, and José R. Paramá. An efficient compression code for text databases. In *25th European Conference on IR Research, ECIR 2003; LNCS 2633*, pages 468–481, Pisa, Italy, 2003.
15. M. Burrows and D. J. Wheeler. A block-sorting lossless data compression algorithm. Technical Report 124, 1994.
16. John Carpinelli, Alistair Moffat, Radford Neal, Wayne Salamonsen, Lang Stuiver, Andrew Turpin, and Ian Witten. Word, character, integer, and bit based compression using arithmetic coding, 1999.
17. Philip Gage. A new algorithm for data compression. *C Users Journal*, 12(2):23–??, February 1994.
18. D. A. Huffman. A method for the construction of minimum-redundancy codes. In *Proc. Inst. Radio Eng.*, pages 1098–1101, September 1952. Published as Proc. Inst. Radio Eng., volume 40, number 9.
19. Snyder J.P. *Map Projections - A Working Manual.* U.S. Geological Survey Professional Paper 1395, United States Goverment Priting Office, 1987.
20. A. Moffat. Word-based text compression. *Software - Practice and Experience*, 19(2):185–198, 1989.
21. Edleno Silva de Moura, Gonzalo Navarro, Nivio Ziviani, and Ricardo Baeza-Yates. Fast and flexible word searching on compressed text. *ACM Transactions on Information Systems*, 18(2):113–139, April 2000.
22. Gonzalo Navarro, Edleno Silva de Moura, M. Neubert, Nivio Ziviani, and Ricardo Baeza-Yates. Adding compression to block addressing inverted indexes. *Information Retrieval*, 3(1):49–77, 2000.
23. Rigaux P., Scholl M., and Voisard A. *Spatial Databases: with application to GIS.* Morgan Kaufmann Publishers, Academic press, 2002.
24. Burrough P.A. and McDonnell R.A. *Principles of Geographical Information Systems.* Oxford University Press, 1998.
25. Inc. Vivid Solutions. The Jump Project. Available at http://www.jump-project.org, 2005.
26. George K. Zipf. *Human Behavior and the Principle of Least Effort.* Addison-Wesley (Reading MA), 1949.
27. Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, 1977.

Networking

Entity Management and Security in P2P Grid Framework

T. N. Ellahi, B. Hudzia, L. McDermott, T. Kechadi, A. Ottewill

Parallel Computational Research Group,
School of Computer Science and Informatics,
University College Dublin, Belfield, Dublin 4, Ireland
tariq.ellahi@ucd.ie, benoit.hudzia@ucd.ie
liam.mcdermott@ucd.ie, tahar.kechadi@ucd.ie, Adrian.Ottewill@ucd.ie

Abstract. During the last decade there has been a huge interest in Grid technologies, and numerous Grid projects have been initiated with various visions of the Grid. While all these visions have the same goal of resource sharing, they differ in the functionality that a Grid supports, characterization, programming environments, etc. In this paper we present a new Grid system dedicated to deal with data issues, called DGET (Data Grid Environment and Tools). DGET is characterized by its peer-to-peer communication system and entity-based architecture, therefore, taking advantage of the main functionality of both systems; P2P and Grid. DGET is currently under development and a prototype implementing the main components is in its first phase of testing. In this paper we gives description of two main components of DGET: Entity Management and Security subsystem.

1 Introduction

In recent years, Internet-scale systems have been developed and deployed to share resources at a very large scale across the traditional organisational boundaries. The need for constructing such systems was motivated by the increasingly complex requirements of modern applications from diverse disciplines. Such global scale systems provide opportunities to harness idle resources which are distributed and heterogeneous. Another benefit offered by such systems is that they allow coordinated use of resources from multiple organisations. Thus, these wide-area systems may span multiple organisations and form virtual organisations on top of the existing organisational hierarchies. Two such systems exploiting these views include Grid and Peer-to-Peer (P2P) systems. Grid and P2P have seen a rapid evolution and widespread deployment. The two technologies appear to have the same final objective, pooling and coordinating large sets of distributed resources[1]. During the last few years various projects have been undertaken to try to merge these two complementary approaches of these technologies, such as OurGrid[2]. Also various modifications to the Globus toolkit[3] have been proposed to include P2P technology and thus improving the discovery system[4].

© A. Gelbukh, S. Torres, I. López (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 19, 2006, pp. 121-136

Typically, Grid systems are designed to run applications with intensive computing and storage needs across the traditional organisational boundaries[5–7]. They are characterised by their sophisticated resource management and data transfer components. P2P systems on the other hand were mainly designed for resource sharing, mostly files. Therefore, the focus of P2P systems is on providing sophisticated resource discovery capabilities. Both approaches have their own advantages and disadvantages.

In this paper we describe DGET (Data Grid Environment & Tools). DGET is a P2P based grid middleware. This paper explains the functionality of two main components of the middleware: EntityManager and Security. Details of DGET architecture and other components can be found in [8][9][10][11]. The rest of the paper is structured as follows: Related work is described in section 2. Section 3 and 4 explain general overview and an intrduction to DGET architecture respectively. Details about Entity Management are given in section 5 and Security subsystem is explained in section 6. The paper concludes in section 7.

2 Related Work

DGET is P2P Grid middleware and employs techniques from both fields. DGET should be compared to other midlewares adopting the P2P Grid approach. The following paragraphs describe how DGET is distinguished from existing solutions.

DGET and Grid Middleware: A number of grid midleware has been developed and used. A wide range of systems have been developed. Some of these focus on providing the core middleware services while other programming frameworks are built on top of these middleware systems and provide high-level application development functionalities. Globus, Legion and UNICORE are the most notable grid middlewares. The Globus Toolkit is the most widely used middleware. DGET has some distinct characteristics. First, existing grid middlewares adopt a manual and static topology whereas DGET is based on dynamic, self-organizing topology borrowed from the decentralised P2P systems. Other distinguishing DGET features include a decentralized P2P style resource discovery and fine grained access control. Existing grid systems depend on specialized central servers to maintain information about shared resources. DGET, on the other hand adopts the P2P style decentralized resource discovery approach and thus doesn't rely on any specialized servers.

On the security front, Globus possess an extremely powerful security system but it has considerable management overhead. All the users are required to have individual accounts on the machines before they can use the resource. This situation is applicable if there are a limited number of participants. In a situation where a very large number of users are present this technique would become very cumbersome. DGET on the contrary doesn't require users to have individual user accounts on the resources. DGET's security mechanism is based on an extended Java security model. Other aspects where DGET security differs from Globus are the fine-grained access control policies and the resource quota control. DGET uses XACML[12] to define fine-grained access control policies.

DGET and Hybrid Systems: Some system designers have tried integrating both P2P and Grid approaches to come up with a system which enjoys the benefits of both grid and P2P systems[4]. This section compares DGET's approach with such hybrid P2P Grid systems. Our Grid is one such P2P Grid middleware. OurGrid[2] bears many similarities with DGET but has some differences as well. Our Grid lacks sophisticated resource discovery solutions present in DGET. Another difference between DGET and OurGrid is migration support. DGET supports strong transparent migration of applications but OurGrid does not.

3 DGET Overview

As described in the related work section, there are a few systems that have combined the concepts from both Grid and P2P systems. Such hybrid systems are called P2P Grids. DGET adopts the same approach and exploits the advantages of both systems and provides an integrated environment for manipulating and analyzing very large data sets.

3.1 DGET Objectives

We have set the following high-level objectives for DGET middleware.

Uniform Management Interface: Resources in DGET systems are represented through a standard and uniform interface. This approach helps in masking the intra-resource heterogeneity. Users don't have to master the entire heterogeneous interface. New resources can be seamlessly added to the system.

Simplicity & Ease of Use: Grid users are typically non-technical, therefore, it is imperative that grid middleware should be simple and easy to use. DGET should tackle the low-level complexities and make it simple for the grid users to use and manage.

Fault Tolerance: In a large scale grid system, faults are not an exception but a norm. DGET should be able to manage the survive system failures transparently without degrading the application performance.

Scalable Architecture: DGET architecture should be scalable to accommodate thousands or even millions of users, resources and data sets. DGET topology must be decentralized and dynamic as centralized architecture result in poor scalability of the system

3.2 DGET Concepts

Entity: An Entity is a network enabled discrete unit of abstraction that provides some functionality to its users. Entity can take many forms e.g. a remote computation, a remote object, a server that processes user requests etc. The Concept of an entity is akin to a process in the operating systems. An Entity is a mobile element that can move around on different nuclei. An Entity is composed of two parts, a system provided Shell and user provided Ghost. Definitions of these are given below.

Shell The Shell is the system provided control part of the entity. Shell exposes a management interface through which entities can be manipulated. Shell is attached to the programmer provided Ghost when an entity is created.

Ghost The Ghost represents the programmer provided part of an entity. Ghost implements the actual logic of the functionality.

Nucleus The Nucleus is the kernel of the system. It Provides basic services like lifecycle management, communication, security etc. to entities.

Connector Transport protocol agnostic communication medium provided to entities for communicating with each other. Connector is a polymorphic artifact that supports a rich set of interaction models between the entities. Connector is a high level construct which shields programmers from low-level connection setup related operations. Another distinguishing feature of connector is that it is a restorable communication medium which plays a key role in the entity migration process.

4 DGET Architecture

In this section we will give an overview of the architectural components. Detailed description of these components is given in their respective sections. The purpose of this section is to give an overview about how all the components are structured and organised. Figure 1 shows a diagrammatic overview of the system. The DGET system is composed of three logical layers. The Following is a brief description each layer and the components residing in that layer.

Core Layer This layer provides basic services to the entities executing in the nucleus. These basic services include communication facilities, lifecycle management and security.

Facilitation Layer This is the second layer in the system. It facilitates execution of the entities by providing them certain services. The components residing at this layer are also modeled as entities. Entities residing at this layer are called System entities. System entities use the services provided by the core layer. Certain components from the Core layer are modeled as system entities as well. Therefore, in the diagram, Security and EntityManager components span both Core and Facilitation layers. The following entities are located at this layer.

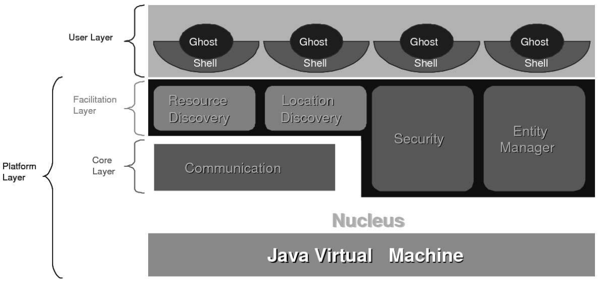


Fig. 1. DGET Architectural Compenents

- **Entity Manager Entity** This entity provides lifecycle management services. This entity to instantiate new entities or manipulate existing entities.
- **Policy Entity** This entity serves as the policy repository of the nucleus. Access control and other management related policies are maintained in the policy entity.
- **Resource Discovery Entity** This entity implements the DGET resource discovery component. Resources and services provides by other entities are discovered through resource discovery entity.

User Layer This is the top most layer in the system. Entities developed by the users and deployed into the system reside at this layer. Entities located at this layer provide user implemented functionality to the users.

5 Entity Management

Entity Creation & Isolation The `EntityManager` entity is responsible for initiating the creation of a user entity in the Nucleus. As described previously, the `EntityManager` functionality is exposed as a System Entity in the Nucleus. The `EntityManager` Entity (EME) publishes its existence along with the characteristics of the host so other entities can locate EMEs according to their requirements. In order to access the local EME running in the same Nucleus, entities can use `EntityContext`. EME creates a shell and passes it the system parameters required to load a ghost. These system parameters include a `GhostLoader` reference, `ThreadGroup` reference and information about the Ghost to be instantiated. Shell uses these parameters and instantiates the Ghost. After successful instantiation of the Ghost, the Shell calls the `setEntityContext()` method on the Ghost class passing in the `EntityContext` object. The Shell also passes an instance of itself to the Ghost. The Ghost can use this instance to invoke lifecycle management operations on itself. The `EntityContext` class is the medium ghosts can use to access system services supported by the Nucleus.

```
public class EntityContext {
  public Connector getEMEntity();
  public Connector getRDEntity(){};
  public Shell getShell();
  public NucleusInfo getNucleusInfo();
  public Resource[] getResourceLimits();
  public Resource[] getResourceConsumption();
}
```

Entity isolation in the Nucleus is provided by a custom classloader called `GhostLoader`. A separate GhostLoader is used to load all the classes belonging to an entity thus providing a separate namespace for the entity classes. GhostLoader associates a security context with the entities classes. This security context is used during its execution to take the access control decisions. Communication between entities is done through connectors. `GhostLoader` has other functions in DGET beside providing separate namespaces for entity classes. These include instrumenting entity bytecode to support soft termination, transparent migration and resource control etc.

Soft Termination Entity termination means killing all the threads an entity might have created during its execution. Sun has declared the thread termination methods as potentially unsafe[13] and deprecated them. Another approach should be adopted to terminate all the threads belonging to an entity. DGET uses the following approach for soft termination of entity threads:
An `Execution` class is introduced. This `Execution` class has a flag indicating the execution state of the entity. During the execution, all entity threads call the `check()` method periodically. If execution flag is RUNNING, `check()` method returns silently but if execution state is TERMINATED, the `check()` method throws
`EntityTerminatedException`. During the loading process, entity classes are instrumented with this execution checks. All the methods are also instrumented with `try/catch` blocks. The `catch` catches the `EntityTerminatedException` exception and re-throws this exception to propagate it down the thread stack. Entity classes are not allowed to catch this exception. During the classloading and instrumentation process, entity class files are scanned to find exception handlers for the `EntityTerminatedException`. This scanning ensures the malicious programmer don't catch this error in order to avoid the entity termination.

5.1 Migration Support

One of the distinguishing features of DGET is strong migration support in a transparent manner. This section describes implementation details of migration support in DGET.

Implementation Techniques The deciding factor in choosing these methodologies were the requirements of portability of the solution, minimal space and time overhead. Code blocks injected into entity classes through bytecode unstrumentation perform different functions like program counter restoration and execution checkpoints (described shortly). The bytecode instrumentation is performed at class load time by a custom classloader. Bytecode instrumentation is performed by the classloader using the Byte Code Enginerring Library(BCEL)[14]. The Second technique used for capturing and restoring execution state is the Java Platform Debugger Architecture (JPDA). JPDA is part of the JVM specification and thus it is implemented by every standard JVM implementation. JPDA provides access to runtime information of the JVM including the thread stacks. JPDA is implemented purely in Java so our migration solution doesn't lose portability

Migration Enabling Features These paragraphs explain the features that enable transparent strong migration in DGET. In order to perform migration at an arbitrary point, values on the operand stack must be saved and restored during the entity restoration process. JPDA doesn't expose any methods to access the values currently present on the operand stack. Initiating migration at such point might result in loss of data from the operand stack. One solution could be to insert checkpoints in the code at locations where execution is not in the middle of a source code level instruction. Migration requests should be delayed till the execution reaches any such checkpoint. Execution checkpoints are inserted as the first instruction in every method and in all the loops in every method.

Another DGET feature to support multi-threaded migration is Mobile Monitors. Java provides multi-threading support in the form of `synchronized` methods and code blocks. A monitor is associated with each java object by JVM and before entering a `synchronized` method or code block, a thread has to acquire the monitor associated with the object. Monitors associated with java objects are maintained and hidden inside the JVM. These monitors are not serilizable and thus are not transported with the serialized objects. Mobile monitors preserve the lock state upon migration. During the class loading process, class files are instrumented to replace Synchronized methods and code blocks. A Mobile monitor is associated with a class that requires synchronized access.

Migration Process

Entity Suspension: The migration process is initiated when the `export()` method is invoked on the Shell. Execution checkpoints discussed in the previous section are used to halt the execution of the entity The `export` method calls the `suspend()` method on the associated `Execution` class. As a result, execution of all the threads is blocked on the next execution checkpoint.

State Capture The `StackFrame` class from JPDA represents a method call on the thread stack. The `StackFrame` class gives access to the values of local variables and the program counter. Calling the `visibleVariable()` method on the `StackFrame` class returns a list of all the variables accessible till the point of execution in the method code. Execution state of all the entity threads along with the mobile monitors and Execution class is saved in a serializable format and transported to the destination for reincarnation of the entity.

State Restoration On the destination nucleus, entity state is restored by calling the `import()` method of the shell. Saved image of entity's execution context is passed as a parameter to the `import()` method.To reestablish the execution state of a thread, its method stack must be rebuilt. To do this, all the methods are called in the order they were on the stack before execution was suspended and migration was initiated. Event handlers can be set that are called when method entry/exit event occurs. When a method entry event occurs, such event handlers restore the values of local variable of the method from the saved execution image. After restoring local variables execution should continue from the code position which is the method invocation for the next method on the stack. Doing so will ensure the instructions already executed are skipped and restoration of the next method on the stack frame begins and proceeds in the same manner. After restoring all the threads to the state they were before the migration was initiated, the `resume()` method on the `Execution` class is called. This method sets the execution status flag to RUNNING and notifies all the threads blocked on this class. Execution proceeds normally afterwards.

As mentioned in the previous paragraph, after restoring local variables, execution should continue from the code position which is the method invocation of the next method on the stack. No mechanism is available in JPDA to set the value of the program counter register to this code position. This problem is solved by maintaining an artificial program counter (APC) which represents an index of method invocations in the method. This APC is incremented after every method invocation instruction. This APC is used in conjunction with a `tableswitch` bytecode instruction which branches the execution according the value of the APC. This `tableswitch` and APC increment instructions are added during the instrumentation procedure. `tableswitch` is added at the beginning of each method and defaults to the original starting code position of the method code.

6 Security

In opposition to grid systems, no centralized servers are present in P2P thus security should not rely on the presence of a centralized server to store and process security related information. All the security related decisions should be made in a decentralized manner making the system scalable. The security model in DGET is designed keeping in mind the P2P system characteristics. The following are the important features of the DGET security model:

- Distributed low-overhead identity based authentication mechanism
- Policy based fine-grained access control
- Distributed security policy management
- Fine grained need-to-act based permission delegation

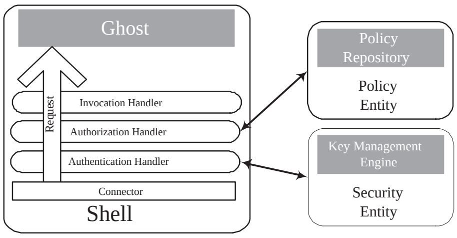


Fig. 2. Security Handlers

6.1 DGET Security Architecture

DGET employs several techniques and components to provide a secure execution environment for entities. This section provides a brief overview of the DGET security architecture. The description of the security system architecture is as follows:

Security Policy-Aware Resource Discovery DGET is equipped with a sophisticated P2P style resource discovery system[8][9]. It is important to enhance resource discovery with security policies so that only those resources are discovered which the user has access to thus increasing system efficiency and productivity. Access control policies are advertised along with the resources so the resource discovery system can analyze this policy during the resource discovery process.

Security Handlers The Shell being the control part of an entity, is the most logical place to perform security related operations. The Shell is equipped with a set of security handlers which process the request to apply security functionality. These handlers are structured in the form Chain of Responsibility (CoR) design pattern. The following two handlers perform security operations:

Authentication Handler This handler performs the authentication and establishes identity/attributes which can be used during the authorization decision process. Details of the authentication mechanism are described in the following sections

Authorization Handler After the successful completion of the authentication process by the authentication handler, user attributes are extracted from user credentials and the request is passed on to the Authorization Handler (AH). This handler carries out the authorization decision process.

Policy Repository Access control information is specified using policies which are updated dynamically. There are multiple levels of security polices specified by different users according to their roles in the system. Policy information is maintained in a separate Entity called the Policy Entity. The Policy Entity is a System Entity and thus it is also subject to the authorization.

Security Entity Security Entity is responsible for the creation and verification of certificates, keys and signatures. The SE is considered as a Trusted Authority (TA) that is valid PKG with the necessary information proving its validity. In the current implementation of DGET, we have used a hard coded certificate that exists in all nuclei to ensure a high level of security for the system. The SE plays the role of Keystore for all the other Entities within the Nucleus Since this Entity holds the keys and handles signature verification, we implemented a cache system for frequent authentication and signature verification.

6.2 Authentication Mechanisms

We have designed Identity Based Cryptographic (IBC) solution to handle the authentication of DGET. We were inspired by various solutions that appeared lately in[15, 16, **?**]. It provides an easy way to manage keys and the benefits from the ID-based approach include:

- Automatic revocation via expiry of time-limited identifiers.
- Reduced round trips if the user can predict delegation requests.
- Reduced bandwidth.
- Similar computational costs.
- Trivial computation of proxy key pairs (RSA key pair generation replaced by elliptic curve multiplication).
- Replication of existing GRID security capabilities.
- Possibility of providing Signencryption scheme.

The IBC system we decided to implement is a variation of the SOK-IBS[ref]

SOK-IBS scheme: This subsection gives formal definitions of presumed hard computational problems on which the SOK-IBS relies.

Bilinear Maps Let G be a cyclic additive group generated by P, whose order is a prime q. Let V be a cyclic multiplicative group of the same order. We use Weil or the Tate pairing ($\hat{e} : G$ x $G \rightarrow V$) over supersingular elliptic curves or Abelian varieties over finite field because they can provide admissible maps over cyclic groups satisfying those properties[17]:

- Bilinearity:
  For any $P,\ Q,R \in G$, we have $\hat{e}(P + Q,R) = \hat{e}(P\ ,R)\hat{e}(Q\ ,R)$ and $\hat{e}(P,Q + R) = \hat{e}(P,Q)\hat{e}(P,R)$.
  In particular, for any a, b $\in \mathbf{Z}_q$, $\hat{e}(aP,\ bP\ ) = \hat{e}(P,\ P)^{ab} = \hat{e}(P,\ abP\ ) = \hat{e}(abP,\ P)$.
- Non-degeneracy:
  There exists $P,Q \in G$, such that $\hat{e}(P,Q) \neq 1$.
- Computability: There is an efficient algorithm to compute $\hat{e}(P,Q)$ for all $P,Q \in G$.

Diffie-Hellman problems Consider a cyclic group $\mathbf{G}_1$ of prime order q.

- The **Computational Diffie-Hellman problem** (CDH) in $\mathbf{G}_1$ is, given $\langle$ *hP, aP, bP* $\rangle$ for un- known *a, b* $\in \mathbf{Z}_q$, to compute $abP \in \mathbf{G}_1$.
- The **one more CDH problem** (*1m-CDH*) is, given $\langle P,\ aP\ \rangle \in \mathbf{G}_1$ for an unknown $a \in \mathbf{Z}_q$, and access to a target oracle[18] $T_{\mathbf{G}_1}$ returning randomly chosen elements Y_i *in* $\mathbf{G}_1$ (*for i = 1; .... ;* q_t, q_t) being the exact number of queries to this oracle) as well as a multiplication oracle.
  $H_{\mathbf{G}_{1,a}}(.)$ answering aW $in\mathbf{G}_1$ when queried on an input W $in\mathbf{G}_1$, to produce a list $((Z_1;\ j_1),\ \ldots\ ,\ (Z_{qt}\ ,\ j_{qt}))$ of q_t pairs such that $Z_i\ =\ aY_{ji}\ in\mathbf{G}_1$ for all i = *1,...,* q_t, $1 \leq j_i \leq q_t$ and $q_m < q_t$ where qm denotes the number of queries made to the multiplication oracle.

Scheme The modified SOK-IBS scheme was proven to be as secure as the one-more Diffie-Hellman problem [19]. This scheme is made of four operations:

- Setup:Given a security parameter *k*, the Private Key Generator (PKG) selects groups G_1 and G_2 of prime order $q > 2^k$, a generator P of G_1, a randomly chosen master key s $\in Z_q$ and the associated public key P_{pub} = *sP*. It also selects cryptographic hash functions of the same domain and range H_1,H_2 : 0, 1 → G_1*. The public parameters of the system are:
  params = (G_1 , G_2 ê, P, Ppub,H_1,H_2)
- KeyGen:Given the ID of a user, the PKG computes
  Q_{ID} = H_1(ID) $\in G_1$ and the associated private key
  $d_{ID} = sQ_{ID} \in G_1$ that is transmitted to the user.
- Sign:In order to sign a message *M*,
  - Pick a random integer $r \in Z_q$ and compute
    $U = rP \in G_1$. Then $H = H_2(ID,M,U) \in G_1$.
  - Compute V = d_{ID} + rH $\in G_1$ where + indicates addition operation on the group G_1.

  The signature on M is the pair = (U, V) $\in G_1$ x G_1.

- Verify:To verify a signature = (U, V) $\in G_1$ x G_1 on a message M, the verifier first obtains the ID of the signer and computes Q_{ID} = H_1(ID) $\in G_1$. The verifier recalculates H = $H_2(ID,M,U) \in G_1$.
  The signature is accepted if ê(P, V) = ê(P_{pub},Q_{ID})ê(U,H) and is rejected otherwise.

The multi-authority scalable DGET Authentication systems: As noted in [19] the use of a random seed to handle unlinkability concerns allows the sender and receiver to have different PKG since no pairing is involved with the receiver's private key (and respectively with the sender's one). This allows us to provide signature and hence authentication capabilities as long as the public keys of the involved PKGs are trusted. Such specific functionality is used to create a hybrid solution between a full identity based solution like a hierarchical ID based one and traditional PKI system. This solution allows more flexibility than traditional PKI or HIBE while reducing the overall network load and computational overhead on the system. In the next section we will describe how we implemented such a system.

Every single Entity, Nucleus and user has its own ID. So before becoming part of DGET every element must register its identity with a TA. Upon a successful registration of ID, the TA will issue the corresponding private key. Every Nucleus possess a Security Entity, so no remote secure communication channel is required to be open and the authentication process is purely local. Since the public key is simply the identity string, this allows a more fine grained control of the key management by adding more information to the identity string such as: a validity period ,delegation characteristic, security domain restriction,etc... The validity period depends on the type of element registered. While a Nucleus might get a long validity period an Entity will get a shorter one corresponding to their average lifecycle. This means that in some cases the authentication string will need to be renewed due to an excessively short validity period.

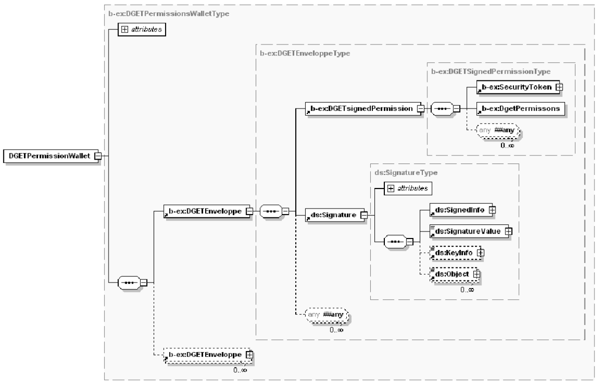


Fig. 3. Entity permissions set Schema

6.3 Policy Repository

Every Nucleus has its own Policy Entity storing its local policies independent of other participants in the grid community. The Policy Entity exposes interfaces to insert, update and delete policies, making the policy administration easier. Policy updates become immediately visible to the authorization decision process. The Policy Entity can be configured to retrieve policies from external sources as well. DGET supports multiple levels of policies governing different aspects of the system. Different policy levels supported by DGET are described below:

Domain Policy Domain policies specify the access control rules defined according to the domain in which the Nucleus is running. Domain policies can be either specified or the Policy Entity can be configured to retrieve a Domain policy from the domain policy repositories.

Nucleus Policy Nucleus policy specifies the policies governing access to system resources and entities. Nucleus policies are specified by Nucleus administrators.

Entity Policy Entity policies decouple access control logic from the Entity application logic and thus updates can be made without changing the Entity code or without redeploying the Entities. Entity policy typically specifies which operations on the Entity are allowed by which users or Entities.

Acting Policy The Acting policy is used for fine-grained permission delegation. The Acting policy follows the least privilege principle and thus allows users to allocate fine grained permissions on the need-to-act basis. The Acting policies controls the level of permissions granted to tasks or requests.

6.4 Authorization and Access Control

Access to scarce system resources and Entities must be controlled and subject to verification of permissions on the resource or perform an operation on the Entity.

Authorization Model DGET uses an association based authorization model. Permissions are organized as authorization profiles. Permissions are granted to profiles and members from the P2P groups, depending on the agreements between the members. Users are required to present credentials to prove their membership with any organization. Based on the membership credentials presented, permissions from the corresponding authorization profiles are granted to the users.

Permission Delegation DGET provides functionality to support fine grained permission delegation following the least privilege principal. Entities can be granted permissions on a need-to-act basis thus avoiding any potential security problem. Users can delegate permissions temporarily to other users who are not members of the organization, or who have rights granted to it. Short-term ad-hoc collaboration scenarios can be supported with this feature. These delegated permissions are attached to the request as the Acting Policy and is evaluated as the part of authorization decision process.

Access Control The Grid is an inherently insecure environment because code is remotely downloaded and executed. Access to shared resources and services must be controlled in order to avoid any misuse. Access control must be exercised at two levels: Nucleus and Entity. The following two paragraphs explain these two aspects of access control:

Nucleus Protection The Nucleus provides the execution environment for Entities. During execution, Entities access the system resources like memory, disk and network etc. It is of utmost importance that access to these resources is controlled. In the absence of such protection mechanisms, some malicious Entities can hijack the Nucleus thus resulting in a Denial of Service (DoS) attack. Nucleus protection is achieved through the Java sandboxing mechanism. DGET uses customized classloaders called GhostLoaders to load ghost classes. GhostLoaders associate appropriate ProtectionDomains based on the authorization profiles.

Entity Protection Besides Nucleus protection, Entities running inside the Nucleus should be protected from misuse as well. Entity owners specify Entity access control policies. These policies are put in place during the deployment. Method invocations on Entities are intercepted and processed through the Authorization Handler. If the method invocation is permitted , the invocation request is processed, otherwise it is rejected.

7 Conclusion

This paper described two main components of the DGET architecture. DGET simplifies the deployment, management and usage of grid systems. DGET provides a dynamic and scalable solution for entity management and security operations that relies on truly decentralized and self-organizing topology. DGET enables resource sharing with the least management overhead and makes grid programming an easier task. DGET provides a flexible interface to adopt any security model. In the future, we plan to incorporate more sophisticated features like fine-grained resource control thus making it feasible to provide Quality of Service (QoS)and support and enforce Service Level Agreements (SLA).

References

1. Adriana Iamnitchi Ian Foster. On death, taxes, and the convergence of peer-to-peer and grid computing. In *2nd International Workshop on Peer-to-Peer Systems (IPTPS'03)*, 2003.
2. G. Germoglio N. Andrade, L. Costa and W. Cirne. Peer-to-peer grid computing with the ourgrid community. In *Proceedings of the SBRC 2005 - IV Salão de Ferramentas (23rd Brazilian Symposium on Computer Networks - IV Special Tools Session)*, May 2005.

3. I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(2):115–128, Summer 1997.
4. Domenico Talia and Paolo Trunfio. Toward a synergy between p2p and grids. *IEEE Internet Computing*, 7(4):96–95, 2003.
5. Ian Foster. The anatomy of the Grid: Enabling scalable virtual organizations. *Lecture Notes in Computer Science*, 2150:1–??, 2001.
6. I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The physiology of the grid: An open grid services architecture for distributed systems integration. *Open Grid Service Infrastructure WG, Global Grid Forum*, 2002.
7. A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke. The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. *Journal of Network and Computer Applications*, 23:187–200, 2001.
8. Adrian Ottewill Benoit Hudzia, M-Tahar Kechadi. Treep: A tree based p2p network architecture. In *IEEE Internatonal Conference on Cluster Computing (Cluster 2005)*, 2005.
9. T.N. Ellahi and M-T. Kechadi. Distributed resource discovery in wide area grid environments. In *The 1st International Workshop on Active and Programmable Grids Architectures and Components APGAC'04, Krakow, Poland*, 2004.
10. T.N. Ellahi B. Hudzia, L. McDermott and T. Kechadi. Entity based peer to peer in data grid environments. In *17th IMACS World Congress, Paris, France*, 2005.
11. T.N. Ellahi B. Hudzia, L. McDermott and T. Kechadi. A java based architecture of p2p-grid middleware. In *The 2006 International Conference on Parallel and Distributed Processing Techniques and Applications*, 2006.
12. Time Moses. extensible access control markup language (xacml) version 2.0. In *OASIS Standard*, February 2005.
13. Sun Microsystems Inc. Why are thread.stop, thread.suspend, thread.resume and runtime.runfinalizersonexit deprecated? http://java.sun.com/j2se/1.4.2/docs/guide/misc/ threadprimitivedeprecation.html (visited 16-jan-06).
14. The byte code engineering library (bcel) http://jakarta.apache.org/bcel/ (visited 16-jan-06).
15. Webno Mao. An identity-based non-interactive authentication framework for computational grids. Technical report, Trusted System Laboratory, HP Laboratories, June 2004.
16. H.W. Lim and K.G. Paterson. Identity-based cryptography for grid security. In *Proceedings of the 1st IEEE International Conference on e-Science and Grid Computing (e-Science 2005), Melbourne, Australia*, 2005.
17. Ian F. Blake, G. Seroussi, and N. P. Smart. *Elliptic curves in cryptography.* Cambridge University Press, New York, NY, USA, 1999.
18. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
19. B. Libert and J. Quisquater. The exact security of an identity based signature and its applications, 2004.

© A. Gelbukh, S. Torres, I. López (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 19, 2006, pp. 137-146

Design to Improve S*3 for a Multilayer-Switched Network in an Institution

Meenakshi Sundaram.K[1], Karthik.B[2], Harihara Gopalan.S[3]

1. Lecturer ,Sri Ramakrishna Engineering College ,
Coimbatore . India .
meenaksji@gmail.com

2. PG Scholar, IGNOU,
New Delhi. India .
karthikbellan@yahoo.com

3. PG Scholar, PSG College of Arts&Science,
Coimbatore. India .
urshari@gmail.com

Abstract. The design and implementation of structured computer and communication network are based on the requirement of an individual, feasibility and application services planned on the network. Modification in the network infrastructure if any must minimize the changes in the network infrastructure with minimum down time. Layered network design attracts most of the organization due to its adoptability, security and scalability. In the layered approach, the fault identification is simple. By implementing Virtual LAN (VLAN) one can suppress the broadcasting, implement access list for security and slice the bandwidth based on application. Security measures can also be considered into the design without much modification by implementing the local security policy. We propose a layered network design for an academic institution to cover entire campus with high-speed data, QoS on multimedia information as well for video lectures with scalability and security aspects.

Keywords: Switched Network , *Scalability ,* Securability ,*Service

1 Introduction

Information networks have emerged as strategic [6] assets and a critical element for delivering education and services. Today's information networks must meet increasing demands to carry more information and provide new services. Leading educational institutions are adopting newer applications for education and information dissemination. The educational institutions participate in research and development activities in addition to the conventional teaching to keep ties with the industry and this leads driving forces for the improvement of network infrastructure and

© A. Gelbukh, S. Torres, I. López (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 19, 2006, pp. 147-156

technology decisions. In educational institutions, the computers are connected with Local Area Network (LAN) technology and (WAN) Wide Area Network for their various Internet applications and Wireless LAN and WAN [11] for research activities. To be success in R&D activities students are required to refer the electronic form of recent literature, journal of referred. Network access from student hostels and academic departments will promote self-education and learning. The purpose of this paper is to describe the proposed multilayer-switched [2] infrastructure for networking the entire campus [1] to improve network service and security [4]. For smooth running of Institute Network some local security policies and practices must be implemented. The paper also describes security considerations while rapid access to various forms of information to the student community. The proposed design includes application considerations as well as technical considerations while designing a converged infrastructure. The network security and system security are two important issues, should be addressed by any academic institution. The Scope of this paper addresses the infrastructure that will be enabled on the IP multi services infrastructure for various forms of network service including wireless WAN adoptability and security to access campus resources and access to Internet.

2 Multilayer-Switched Network Design Architecture

Multilayer-switched network design architecture includes three layers [5] such as

- The backbone (core) layer that provides optimal transport between sites
- The distribution layer that provides policy-based connectivity [8]
- The local-access layer that provides workgroup/user access to the network

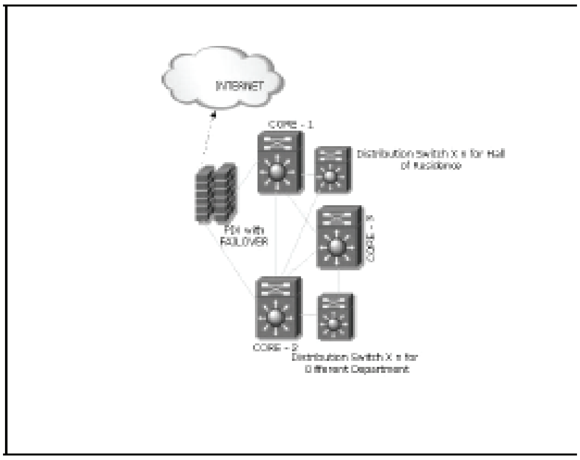

INTERNET
CORE - 1
Distribution Switch X n for Hall of Residence
PIX with FAILOVER
CORE - 3
CORE - 2
Distribution Switch X n for Different Department

Fig 1. A typical multi-layer Switched Backbone

Each student hostel and the Major academic departments will have a high speed layer 3 aggregation switches with local servers. Edge switches, which connect to the student, faculty or lab workstations will aggregate the end connections to the distribution switch. The core switches will provide high-speed transport and switching for the entire campus network infrastructure.

Figure 1, we show the Layer 3 switched campus backbone with dual links to the backbone from each distribution-layer switch. The main advantage of this design is that each distribution-layer switch maintains two equal-cost paths to every destination network, so recovery from any link failure is fast. This design also provides double the trunking capacity into the backbone Layer 3 switched backbones have several advantages such as Reduced router peering, Flexible topology with no spanning-tree loops, Multicast and broadcast control in the backbone and Scalability to arbitrarily large size.

3 Various Layers Of The Multi-Layer Switched Backbone

3.1 Local-Access Layer

It will provides high speed 10 Mbps / 100 Mbps Ethernet connection to the end station / desktop computer on Category 5/6 UTP (copper) cabling. The maximum number of user connections required at the each floor of a wing and is depends on the port density of the switch. It can be easily replaceable in case of failures and is manageable from central location with network management station and supports converged services on IP such as telephony and video. Layer 2 managed switches capable of segregating users based on VLANs (Broadcast domains). The access layer is the point at which local end users are allowed into the network. This layer may also use access lists [10] or filters to further optimize the needs of a particular set of users to enforce the local security policy. In the campus environment, access-layer functions can includes such as shared bandwidth, switched bandwidth, MAC layer filtering and micro segmentation.

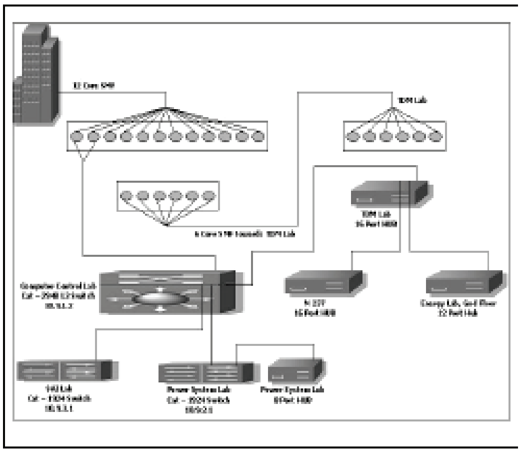


Fig 2. Network Schematic of the Local-Access Layer

3.2 Distribution Layer

It provides high-speed 100 Mbps Ethernet connections / gigabit Ethernet connection [3] to the Department/Hostels on Category 5/6 UTP (copper) cabling. It aggregates all the gigabit Ethernet uplinks on single mode fiber from the access switches at the different areas of the hostels. It is modular and scalable both in terms of expansion and performance. The distribution layer of the network is the demarcation point between the access and core layers and helps to define and differentiate the core. The purpose of this layer is to provide boundary definition and is the place at which packet manipulation can take place. In the campus environment, the distribution layer can include several functions, such as address or area aggregation, departmental or workgroup access, broadcast/multicast domain definition, Virtual LAN (VLAN) routing, any media transitions that need to occur and security. It is manageable from central location with network management station. It supports converged services on IP such as telephony and video, security and access control lists to protect common resources such as network infrastructure switches and critical servers from pilferage and attacks. It can connect to the core switches at the central core switch.

3.3 Backbone (core) Layer

It provides high-speed gigabit switching between the distribution switches. It should be completely redundant. The inter-link between the core switches and the distribution switch will use the single mode fiber infrastructure. It should be manageable from the network management station. It supports converged services on IP such as telephony and video. The core layer is a high-speed switching backbone and should be designed to switch packets as fast as possible. This layer of the network should not perform any packet manipulation; such as access lists and filtering that would slow down the switching of packets. The Campus network is used multilayer-switched Network. Layer 3 switching provides the same advantages as routing in campus network design, with the added performance boost from packet forwarding handled by specialized hardware. Putting Layer 3 switching in the distribution layer and backbone of the campus segments the campus into smaller, more manageable pieces.

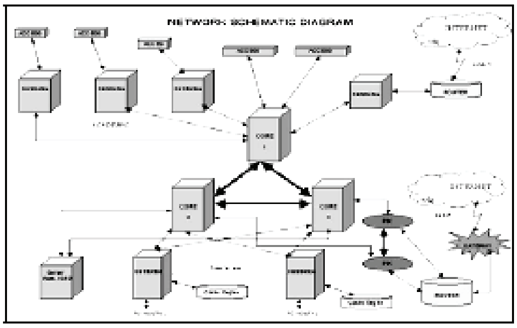


Fig 3. Complete network schematic of the layered network

4 Principles Of The Design Of Layered Network

Good-layered network design [5] is based on many concepts that are summarized by the following key principles:

Examine single points of failure carefully—There should be redundancy in the network so that a single failure does not isolate any portion of the network. There are two aspects of redundancy that need to be considered are backup and load balancing. In the event of a failure in the network, there should be an alternative or backup path. Load balancing occurs when two or more paths to a destination exist and can be utilized depending on the network load. The level of redundancy required in a particular network varies from network to network.

Characterize application and protocol traffic: - The flow of application data will profile client-server interaction and is crucial for efficient resource allocation, such as the number of clients using a particular server or the number of client workstations on a segment.

Analyze bandwidth availability: - There should not be an order of magnitude difference between the different layers of the hierarchical model. It is important to remember that the hierarchical model refers to conceptual layers that provide functionality. The actual demarcation between layers does not have to be a physical link—it can be the backplane of a particular device.

Build networks using a hierarchical or modular model: - The hierarchy allows autonomous segments to be inter-networked together.

5 Multilayer-Switched Infrastructure To Improve Network Service And Security: Implementation Issues

5.1 Reducing the size of Failure Domain

A group of Layer 2 switches connected together is called a Layer 2 switched domain. The Layer 2 switched domain can be considered as a failure domain because mis configured or malfunctioning workstation can introduce errors that will impact or disable the entire domain. A jabbering network interface card (NIC) may flood the entire domain with broadcasts. A workstation with the wrong IP address can become a black hole for packets. Problems of this nature are difficult to localize. Restricting it to a single Layer 2 switch in one wiring closet if possible should reduce the scope of the failure domain.

5.2 Limiting the Size of Broadcast Domain

Media Access Control (MAC)-layer broadcasts flood throughout the Layer 2 switched domain. Use Layer 3 switching in a structured design to reduce the scope of broadcast domains. In order to do this, the deployment of VLANs and VLAN trunking is needed. Ideally one VLAN (IP subnet) is restricted to one wiring-closet switch. The gigabit uplinks from each wiring-closet switch connect directly to routed interfaces on Layer 3 switches.

5.3 Avoidance of Spanning-Tree for Redundancy

Layer 2 switches run spanning-tree protocol to break loops in the Layer 2 topology. If loops are included in the Layer 2 design, then redundant links are put in blocking mode and do not forward traffic. It is preferred to avoid Layer 2 loops by design and have the Layer 3 protocols handle load balancing and redundancy, so that all links are used for traffic. The spanning-tree domain should be kept as simple as possible and loops should be avoided. With loops in the Layer 2 topology, spanning-tree protocol takes between 30 and 50 seconds to converge. Use Layer 3 switching in a structured design to reduce the scope of spanning-tree domains. Let a Layer 3 routing protocol, such as Enhanced Internet Gateway Routing Protocol (IGRP) or Open Shortest Path First (OSPF); handle load balancing, redundancy, and recovery in the backbone.

6 VLAN Design And Configuration

6.1. Perfect VLAN Design

VLAN has the same characteristics of a failure domain, broadcast domain, and spanning-tree domain, as described above. So, although VLANs can be used to segment the campus network logically, deploying pervasive VLANs throughout the campus adds to the complexity. Avoiding loops and restricting one VLAN to a single Layer 2 switch in one wiring closet will minimize the complexity. With the advent of high-performance Layer 3 switching in hardware, the VLANs can be used to logically associate a workgroup with a common access policy as defined by access control lists (ACLs). Similarly, VLANs can be used within a server farm to associate a group of servers with a common access policy as defined by ACLs.

6.2 Configuration of VLAN

When you configure VLANs, the network can take advantage of the following benefits and they are

Broadcast control—Just as switches physically isolate collision domains for attached hosts and only forward traffic out a particular port, VLANs provide logical collision domains that confine broadcast and multicast traffic to the bridging domain.

Security—If you do not include a router in a VLAN, no users outside of that VLAN can communicate with the users in the VLAN and vice versa. This extreme level of security can be highly desirable for certain projects and applications.

Performance—You can assign users that require high-performance networking to their own VLANs. You might, for example, assign an engineer who is testing a multicast application and the servers the engineer uses to a single VLAN. The engineer experiences improved network performance by being on a "dedicated LAN," and the rest of the engineering group experiences improved network performance because the traffic generated by the network-intensive application is isolated to another VLAN.

Network management—Software on the switch allows you to assign users to VLANs and, later, reassign them to another VLAN. Recabling to change connectivity is no longer necessary in the switched LAN environment because network management tools allow you to reconfigure the LAN logically in seconds.

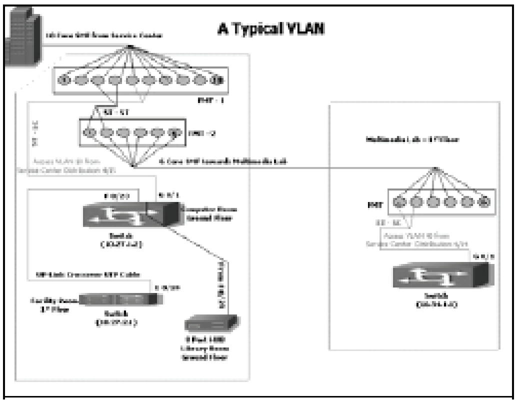


Fig 4. Schematic of a typical VLAN

6.3 IP Subnet Planning for the Campus

An IP subnet also maps to the Layer 2 switched domain; therefore, the IP subnet is the logical Layer 3 equivalent of the VLAN at Layer 2. The IP subnet address is defined at the Layer 3 switch where the Layer 2 switch domain terminates. The advantage of subnetting is that Layer 3 switches exchange summarized reachability information, rather than learning the path to every host in the whole network. Summarization is the key to the scalability benefits of routing protocols, such as Enhanced IGRP and OSPF. In an ideal, highly structured design, one IP subnet maps to a single VLAN, which maps to a single switch in a wiring closet. This design model is somewhat restrictive, but pays huge dividends in simplicity and ease of troubleshooting.

6.4 Policy Domain

Access policy is usually defined on the routers or Layer 3 switches in the campus network. A convenient way to define policy is with ACLs that apply to an IP subnet. Thus, a group of servers with similar access policies can be conveniently grouped together in the same IP subnet and the same VLAN. Other services, such as DHCP are defined on an IP subnet basis.

7 Quality Of Service For Voice And Video And Caching

7.1 Quality of Service for Voice and Video

Interface experiences congestion when it is presented with more traffic than it can handle. Network congestion points are strong candidates for Quality of Service (QoS) mechanisms. A better alternative is to apply congestion management and congestion avoidance at oversubscribed points in the network. Figure 5 shows the examples of typical congestion points

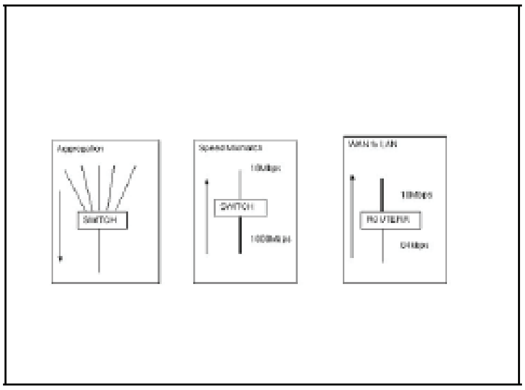


Fig 5. Typical congestion points

Network congestion results in delay. A network and its devices introduce several kinds of delays, as explained in Understanding Delay in Packet Voice Networks. Variation in delay is known as jitter, as explained in Understanding Jitter in Packet Voice Networks. Both delay and jitter need to be controlled and minimized to support real-time and interactive traffic. In particular, QoS features provide better and more predictable network service by different methods such as Supporting dedicated bandwidth, Improving loss characteristics, Avoiding and managing network congestion, Shaping network traffic and Setting traffic priorities across the network.

7.2 Scaling with Caching

As enterprises grow and expand their network over Internet or to remote locations on WAN, it becomes very critical to improve the response to the enterprise web servers and to reduce delay across WAN. Typical Web caching solutions involve a series of caching devices in close proximity to a specific user community. Content Caching works best if cache engines are positioned closest to the points of access to minimize WAN bandwidth utilization. An enterprise can deliver accelerated service to its customers by front-ending Web server farms with cache engine clusters. In this application, content requests are redirected to a cache engine cluster instead of directly forwarding them to the Web servers. If the content being requested is cacheable, the cache engines will fill the request. When the cache cluster fulfills these requests, it off-loads traffic from the Web servers, thereby minimizing content download latency and increasing Web server capacity. Therefore, once a customer requests a particular piece of cacheable content, it is cached so that successive requests are not directed repeatedly to a Web server. The same concept can be extended to the enterprise LAN as well. Intranet servers with rich multimedia content are often the potential bottlenecks. The content can be moved closest to the user community with high-speed cache engines.

8 IP Address Scheme And Network Address Translation

8.1 Private Address Space

The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private Internet [9]
10.0.0.0
10.255.255.255 (10/8 prefix)
172.16.0.0
172.31.255.255 (172.16/12 prefix)
192.168.0.0
192.168.255.255 (192.168/16 prefix)

An institute that decides to use IP addresses out of the address space defined in this document can do so without any coordination with IANA or an Internet registry. The address space can thus be used by many enterprises. Addresses within this private address space will only be unique within the institute, or the set of institute, which choose to cooperate over this space so they may communicate with each other in their own private Internet. Private hosts can communicate with all other hosts inside the institute, both public and private. However, they cannot have IP connectivity to any host outside of the institute. While not having external (outside of the institute) IP connectivity private hosts can still have access to external services via mediating gateways (e.g., application layer gateways). Two scalability challenges facing the Internet are the depletion of registered IP address space and scaling in routing. Network Address Translation (NAT) [7] is a mechanism for conserving registered IP addresses in large networks and simplifying IP addressing management tasks. As its name implies, NAT translates IP addresses within private "internal" networks to "legal" IP addresses for transport over public "external" networks (such as the Internet). Incoming traffic is translated back for delivery within the inside network.

9 Conclusion

We have designed a layered switched network for an academic institution. This design takes provides high-speed data down to the end point. The fault identification is found to be easy. The security policies can be implemented at the VLAN's. VLAN suppresses the broadcasting into the local domain and so we avoid bandwidth choking. The downtime of the core and distribution level is taken care by the redundancy. This design provides effective use of effective IP address space by using private IP addresses and network address translation.

References

1. CISCO AVVID Campus Solution .
   http://www.itworld.com/WhitePapers/Cisco_AVVID_Campus/
2. CISCO Inter-network Design Guide
   http://www.cisco.com/univercd/cc/td/doc/cisintwk/idg4/index.htm
3. Fiber Optics
   http://www.fols.org/pubs/whitepaper0100.html
4. Firewall
   http://www.firewallsdirect.com/white_papers/watchguard/fb_wp_ltr.pdf
5. Gigabit Campus Network Design— Principles and Architecture .
   http://www.cisco.com/warp/public/cc/so/neso/lnso/cpso/gcnd_wp.htm
6. IT Strategic Plan for an academic institute .
   http://athena.uwindsor.ca/units/its/itsp/ITSPWFinal.nsf/
5. University+Website?OpenForm
7. Network address Translation .
   http://www.cisco.com/warp/public/732/nat/
8. Network Connectivity Solutions .
   http://www.intel.com/network/connectivity/solutions/
9. Private IP address .
   http://www.isi.edu/in-notes/rfc1918.txt
10. What do you need for network security?.
    http://business.cisco.com/prod/tree.taf%3Fasset_id=87144&
    public_view=true&kbns=1.html
11. Wireless Wan network .
    http://www.intel.com/network/connectivity/resources/doc_library/documents/
    pdf/np1692-01.pdf

Logic and String Algorithms

A String Metric Based on a One-to-one Greedy Matching Algorithm

Horacio Camacho and Abdellah Salhi

The University of Essex, Colchester CO43SQ, U.K.
{jhcama, as} @essex.ac.uk

Abstract. We introduce a novel string similarity metric based on a matching problem formulation. This formulation combined with other heuristics generates comparatively more accurate string similarity scores than some other methods. The results of the proposed method are improved by training the method on domain data. A detailed description of the method as well as computational results on many databases are given.

1 Introduction

Automatic methods for duplicate record detection such as record linkage, [1], merge/purge, [2], duplicate detection, [3], and Hardening, [4], among others, have been suggested for many years now. Although different in concept, they all require, in one form or other, the use of string similarity metrics, in order to decide if two records are similar enough to be considered as duplicates.

String similarity metrics can be roughly divided into three general groups [5]: Token-based metrics, character-based metrics and hybrid metrics. The token-based metrics, of which Jaccard, [6], Cosine and TFIDF, [7], are members, consider strings as "bags of words", [7]. Character-based metrics such as the Jaro metric, [8], and its variants, count the number of similar characters in a pair of strings. Edit metrics, such as the Levenshtein, [9], and its variants, count the number of character-level operations (delete, insert, substitute) required to transform one string into another treating the string as a sequence of characters. Hybrid metrics combine both the token-based and the character-based metrics. In a hybrid metric, a token-based metric uses scores obtained by a character-based metric. Common examples of hybrid metrics are SoftTFIDF [5], and the metric due to Monge and Elkan [3], also known as Level2 method. For a good survey of string metrics, the reader is advised to consult [5]. There, a comparison between several string metrics has been carried out and SoftTFIDF performed best on average.

Because the results from using individual metrics often lack consistency, techniques such as the Support Vector Machine (SVM) approach, [10], that combines results from different metrics, has been introduced. This regressional type approach may be limited in its applicability due to computational costs particularly when the number of participating metrics is high and the input databases are large. The consistency issue spawned other approaches such as those which rely on training. In [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], and [21] trained techniques have been suggested. Because parameter tuning is done according to what domain the input database is concerned with, consistency in performance is therefore enhanced. So far, however, trained techniques are mainly of the character-based type.

Here we suggest a novel metric for string similarity. The approach is hybrid in nature; it combines a character-based metric and a token-based metric, both explained later. Moreover, it can also be trained for a given domain. The training procedure will be explained later too. Both the non-trained and the trained versions of it are compared on a set of databases with several approaches as can be found in SecondString, [5], and Simmetrics Java toolkits, [22]. The non-trained version performs consistently well in all experiments. But, the trained version performs better and compares favorably with all metrics considered.

In the next section we provide a motivation for this metric. In section 3 we formalize the presented ideas into a model. In section 4 we explain the string metric. In section 5 we illustrate the method with an example. In section 6 we define a method to estimate the parameters of the proposed string metric. Section 7 contains comparison results and section 8, the conclusion.

2 Motivation

Consider a pair of strings T and U, which, through tokenisation, we break into substrings (tokens) as: $T = \{T_1, T_2, ..., T_I\}$ and $U = \{U_1, U_2, ..., U_J\}$. A character-based method calculates all similarity scores $s_{ij}(T_i, U_j)$ between token T_i and token U_j for $i = 1, ..., I$ and $j = 1, ..., J$. Based on these scores, a token-based method selects the most adequate token pairs in order to compute the similarity score $s(T, U)$ of strings T and U. Both scores $s_{ij}(T_i, U_j)$ and $s(T, U)$ take positive and possibly zero values with the large values corresponding to good matches and low values to (potentially) non-matches in both character and token comparisons.

Ideally, similarity metrics (hybrid or otherwise) must be consistent. In other words, the similarity of similar or near similar strings returned must be high and that of “not so similar strings” must be low in comparison. Unfortunately, this is often not the case and the reason may well lie with the way the token similarity is calculated.

Consider the two most common hybrid methods: SoftTFIDF and Level2. SoftTFIDF defines $CLOSE(\theta, T, U)$ as a triplet containing strings T and U and a scalar θ such that for any token T_i included, there is some token U_j such that $s_{ij}(T_i, U_j) > \theta$, and s_{ij} is a similarity score from a character-based string metric, such as Jaro-Winkler, [23]. In contrast, the Level2 method considers the complete set of tokens in T, and then chooses as similar to each token T_i, those tokens U_j from U having: $\max_{j=1}^{J} s_{ij}(T_i, U_j)$ where s_{ij} is a similarity score from a character-based string metric, such as a variant of the Levenshtein method due to Monge and Elkan, [23].

To illustrate what we have just said, we consider the following example: $T = \{\text{John, Johnson}\}$, $U = \{\text{Johns, Charleston}\}$, $V = \{\text{J., Charletson}\}$. SoftTFIDF computes $s\,(T, U) = 0.95$ and $s\,(U, V) = 0.49$. Level2 computes $s\,(T, U) = 1.0$ and $s\,(U, V) = 0.84$. While one might almost be certain that T and U are not pointing to the same real object, and there is more evidence that U and V are more similar than T and U, those methods score $s\,(T, U)$ higher since they consider the same token U_k if all tokens in T are very similar and each token score $s_{1k}, s_{2k}, ..., s_{Ik}$ is very close to 1.

We consider the most appropriate token pairs in a different manner. We define the most appropriate token pairs as a one-to-one matching setting, similar to the following assignment problem:

$$\max z = \sum_{i,j} s_{ij} x_{ij} \tag{1}$$

s.t.

$$\sum_{i} x_{ij} = 1, \forall j \text{ and } \sum_{j} x_{ij} = 1, \forall i \tag{2}$$

$$i = 1, ..., I, j = 1, ..., J, s_{ij} \in R^{+}, x_{ij} \in \{0, 1\}$$

where x_{ij}, is a binary variable which takes value 1 if token T_i and token U_j are considered as a match, and 0 otherwise, but with the difference that here we define matched pairs as the set of highest scored pairs of tokens. In the next section we formalize our method. We also provide an example that explains some drawbacks of the assignment model.

3 Formalization

A one-to-one match between tokens in T and tokens in U implies that the assignment constraints (2) are satisfied. Note that the set of matching pairs we look for are potentially different from those returned when the assignment problem is solved. While the assignment problem maximizes the function (1), subject to restrictions (2), we instead suggest the following procedure:

Algorithm 1: Select a maximum score value $s_{kl} = \max\limits_{ij} s_{ij}$ such that token T_k and token U_l have not been found to match another string yet, or they satisfy: $\sum_i x_{il} = 0$ and $\sum_j x_{kj} = 0$. The token pair (T_k, U_l) is then considered as a matching pair, thus setting $x_{kl} = 1$. This procedure is repeated until restrictions (2) are completely satisfied.

An alternative way to find the set of matching pairs, is by sorting the set of scores s_{ij} in descending order. The pair with maximum score is chosen to be part of the overall match. The process is repeated bearing in mind that restrictions (2) must be satisfied at all time. This process is not computationally expensive and amounts mostly to the cost of a sorting algorithm. In fact it can be reduced further by removing from the subsequent list of scores those which do not correspond to pairs satisfying the restrictions. These are found trivially since all pairs in the row and column of the chosen pairs so far are barred from being chosen by the assignment constraints. This consideration reduces the work substantially.

This differs from the assignment problem in that we always choose those token pairs with a score which is as high as possible for a match, when there is no clear match. For instance, consider the example of Table 1.

Table 1. Example of two similar strings: T and U

String T	Strng U
Gerardo F. Jolmes Dorado	Francisco D. Jolmes Dorado

Let: $s_{14} = 0.6, s_{21} = 0.6, s_{33} = 1.0, s_{42} = 0.6, s_{44} = 1.0$, and all the other scores are set to 0. The solution to the assignment problem is $\{x_{14}, x_{21}, x_{33}, x_{42}\}$ corresponding to the matching pairs {(Gerardo, Dorado), (F., Francisco), (Jolmes, Jolmes), (Dorado, D.)}. Since s_{44} (Dorado, Dorado) has a higher score than s_{42} (Dorado, D.), then the better matching pairs would be {(F., Francisco), (Jolmes, Jolmes), (Dorado, Dorado)} corresponding to the solution $\{x_{21}, x_{33}, x_{44}\}$. This, however, has a lower assignment objective ($z = 2.6$) in (1) than that of the solution returned for the assignment problem ($z = 2.8$). However, we do believe, and this is backed by our results, that in fact this is a better way to settle the matching problem between tokenised strings, in particular when there is no perfect match between the strings. This situation occurs often, since in most databases the ratio of redundant to non-redundant records is more likely to be low than high. It is important to add that when this is not the case then the solution advocated here will return a similar result to that of the solution to the assignment problem.

4 The Hybrid Method

The method about to be suggested is hybrid in nature. In the following we introduce the two aspects of it, the character and the token. As we shall see, some techniques used in the past have also been implemented here, but also some improvements have been introduced. Even though each of the character and the token based metric are defined differently, both apply the same way of assigning pairs (characters and tokens) as defined in Algorithm 1.

4.1 Character-based Similarity Score

Let each token T_i and U_j, $i = 1, ..., I$ and $j = 1, ..., J$, of a pair of strings (T, U), be broken into a set of characters $T_i = \left\{T_i^1, T_i^2, ..., T_i^F\right\}$ and $U_j = \left\{U_j^1, U_j^2, ..., U_j^G\right\}$. Let $\delta\left(T_i^f, U_j^g\right)$, $f = 1, ..., F$, and $g = 1, ..., G$, be a function equal to 1 if characters: $T_i^f = U_j^g$, and 0 otherwise.

Same characters can be found in non similar tokens, for example, from Table 1 the pair of tokens: (Gerardo, Dorado) share the same characters “a”, “r”, “d” and “o”, but the positions of “a” and “r” in this example are making an important difference. In order to account for the order in position of characters, we introduce the following function:

$$d_{ij}^{fg} = \begin{cases} \delta\left(T_i^f, U_j^g\right) - \frac{1}{\gamma}\left|f-g\right|, \text{ if } \left|f-g\right| < \gamma \text{ and } \delta\left(T_i^f, U_j^g\right) = 1 \\ 0, \text{otherwise} \end{cases} \tag{3}$$

where γ is a constant which penalizes the difference between the positions of matching pairs by the quantity: $-\frac{1}{\gamma}\left|f-g\right|$ and at the same time restricts the number of positions where a possible match might be found, if for example, $\gamma = 3$, and if $|f-g| > 3$, then $d_{ij}^{fg} = 0$. This condition also speeds up the computation since no characters are worth exploring when position is bigger than: $|f-g| = 3$.

Given a set of similarity scores of characters d_{ij}^{fg}, by Algorithm 1 we can define a set of matching pairs of characters (T_i^f, U_j^g) by setting $s_{fg} = d_{ij}^{fg}$. Thus, we define our token score as:

$$s_{ij} = \sum_{fg} s_{fg} x_{fg}/(2F) + \sum_{fg} s_{fg} x_{fg}/(2G) \tag{4}$$

where x_{fg} is returned by Algorithm 1.

We can make further improvements to the proposed string metric value s_{ij} by implementing the Winkler scorer [23]. This heuristic has been applied in the past to the Jaro method and is defined as: $s_{ij}^{'} = s_{ij} + prefix * prefixScale * (1 - s_{ij})$, where $prefix$ is the largest prefix which characters of T_i and U_j match, such that a prefix is no larger than 4, $prefixScale$ is a scaling factor which is meant to temper down the upwards adjusted score because of the shared prefix between the strings considered.

4.2 Token-based Similarity Score

Among the token based string metrics discussed in [5], the TFIDF was shown to have the best results. TFIDF [7] is a vector space approach widely used by the information retrieval community, and it has also been implemented in for the task of matching names such as SoftTFIDF. The latter variant has a very good record in name matching. We try here to replicate this success, through a modification of the basic method.

Recall that TF_{T_i} is the frequency of token T_i in T and IDF_{T_i} is the inverse frequency of token T_i in the current dataset or “corpus”. Notice that TFIDF was initially designed for the task of searching documents, here, the searching task is limited to match short strings compound of only few tokens (see Table 1). For example, consider a document which describes the “history of computers”, as it is expected, the token: “computer” will appear several times in the document, in contrast, in our case, the token: “Gerardo” in string T of Table 1, appears only once. Notice that we are also not penalizing the positions of the tokens, as we did in the last section for the positions of the characters, since the number of tokens included into a string is relatively small and repetition of similar tokens is very rare in names.

Here we consider a different way to measure the contribution of the tokens. Instead of measuring the frequency of token T_i (TF-term) we measure a match rate term (MR-term), defined as:

$$MR_{ij} = c_{ij} / \min\left(|T|, |U|\right) \tag{5}$$

where c_{ij} is the number of matched characters of a pair of tokens T_i and U_j and $|T|$ and $|U|$ are the lengths of strings T and U, respectively. For example, from Table 1 in the pair of tokens: (Gerardo, Dorado) $c_{14} = 4$ (“a”, “r”, “d” and “o”), $|T| = 7$, $|U| = 6$ and $MR_{14} = 4/6$.

As in [5] we compute the IDF-term as follows: $IDF_{ij} = a_i a_j$, where: $a_i = b_i / \sqrt{\sum_i b_i^2}$, $b_i = \log\left(IDF_{T_i}\right)$. We then define the set of token scores as:

$$d_{ij} = s_{ij}^{'} \left[\frac{1}{2} IDF_{ij} + \frac{1}{2} MR_{ij}\right] \tag{6}$$

Given a set of similarity scores of tokens d_{ij}, by Algorithm 1 we can define a set of matching pairs of tokens (T_i, U_j) by setting $s_{ij} = d_{ij}$. Thus, we define our string similarity score as:

$$s = \sum_{ij} s_{ij} x_{ij}. \tag{7}$$

where x_{ij} is the solution from Algorithm 1.

5 Example

Consider the pair $T = \{\text{Jhon, Johnson}\}$, $U = \{\text{Johns, Charleston}\}$. We compute the character-based and token-based similarity scores as follows.

By Character-Based String Metric:

In order to compute the score of a token pair, say s_{11} of tokens: (“Jhon”, “Johns”), we partition each token into the set of characterers: $T_1 = \{$“j”, “h”, “o”, “n”$\}$ and $U_j = \{$“j”, “o”, “h”, “n”, “s”$\}$. We arbitrarily set $\gamma = 3$, and the list of character scores d_{ij}^{fg} of (3) is then computed, for instance: $d_{11}^{11} = \delta($“j”, “j”$) - \frac{1}{3}|1-1| = 1$, $d_{11}^{12} = 0$, and repeat this process until all the character scores are computed. Once the list of scores d_{ij}^{fg} is computed, we apply Algorithm 1 in order to select the most appropriate character pairs. The output of Algorithm 1 is the set: $\{x_{11}, x_{23}, x_{32}, x_{44}\}$, and its corresponding scores are shown in Table 2, hence $\sum_{fg} s_{fg} x_{fg} = 3.33$ and $s_{11} = 3.33/8 + 3.33/10 = 0.75$ (see (4)). By the Winkler scorer, we set $prefixScale = 0.1$. Since the only prefix with perfect match is the first character pair (“j”, “j”), $prefix = 1$. Thus: $s_{11}^{'} = 0.75 + 0.1\,(1 - 0.75) = 0.775$.

By Token-Based String Metric:

We compute the scores of all pairs $s_{ij}^{'}$ in the same way as previously illustrated. In this particular case, the IDF term for each token T_i and U_j is the same, since the frequency of each token is equal to 1. The size of the corpus for this small example is 2, thus $IDF_{T_i} = IDF_{U_j} = \ln(2/1)$ and $a_i = a_j = IDF_{T_i}/\sqrt{\sum_i (IDF_{T_i})^2} = 0.707$. We compute the number of matched characters for a given pair of tokens c_{ij}, for instance: $c_{11} = 4$ ("j", "h", "o" and "n"), thus, MR-term is obtained, for instance MR_{11}=$4/\min(11,15)$ (see (5)). The list of scores, MR-terms and the set of scores d_{ij} for all pairs of tokens T_i and U_j are shown in Table 3. By Algorithm 1 we select the most appropriate pair of tokens. The output of Algorithm 1 is the set: $\{x_{21}, x_{12}\}$, hence $s = 0.436{+}0.052 = 0.488$.

Table 2. Character based string metric example of a pair of tokens: ("Jhon","Johns"). The remaining scores are equal to 0

T_1^f	U_1^g	s_{11}^{fg}
T_1^1	U_1^1	1
T_1^2	U_1^3	0.667
T_1^3	U_1^2	0.667
T_1^4	U_1^4	1

6 Parameter Estimation

It is possible to improve the performance of the proposed string metric if we set values of the function $\delta\left(T_i^f, U_j^g\right)$ and the transposition constant γ taking account of the domain of the data. In some cases, a pair of characters might have a chance to be matched if the characters are considered equivalent in the given domain. For example, the characters "-" and "/" might be considered to be equivalent if the data we intend to match a set of telephone numbers. We would then have $\delta($"-", "/"$) = 1$. In other cases similar characters like "e" and "c" might be considered to be similar if the information was extracted via OCR, thus $\delta($"e", "c"$) = 1$.

Given a training set of matching and non matching pairs, finding the equivalent character pairs by hand can be difficult, since the number of possible pairs to tune may be very large.

As in [20], we initially assume independence between matching characters, i.e. a matching character in a pair of tokens is independent of other matching or non matching pairs of characters. If we have n different characters in a given vocabulary and if $\delta\left(T_i^f, U_j^g\right) = \delta\left(U_j^g, T_i^f\right)$, there are is $\frac{n(n-1)}{2}$ combinations of different pairs of characters. If for each corresponding pair: $\delta\left(T_i^f, U_j^g\right)$ can be equal to 1 or 0, then the number increases to $n\,(n-1)$.

Table 3. Token based string metric example of the pair of strings ("Jhon","Johns")

T_i	U_j	s'_{ij}	MR_{ij}	IDF_{ij}	d_{ij}
T_1	U_1	0.775	0.364	0.5	0.335
T_2	U_1	0.914	0.455	0.5	0.436
T_1	U_2	0.175	0.090	0.5	0.052
T_2	U_2	0.121	0.364	0.5	0.052

We can reduce the number of parameters if we define a candidate list of possible matching pairs of characters. Such candidate list can be defined by the algorithm described in [20], where given a set of matching pairs, the probabilities of edit operations of characters such as insertion, deletion and transposition are estimated by an *Expectation Maximization* algorithm, [20].

Once the probabilities are estimated, we ignore the insertion and deletion probabilities and we only consider substitution pairs whose probability is significantly bigger than, for instance, 1×10^{-3}.

Since the independence assumption between characters might not hold in all cases, we set the matching scores iteratively in a greedy manner. We test all the candidate pairs of characters and set as matching those pairs which best improve performance. We repeat this process until no further improvement is achieved. The transposition constant γ can also be iteratively set in the same manner. We consider eleven possible values of γ in the range $(0, 1)$.

The performance we measure in order to find the best parameter is given by the *non-interpolated average precision* as defined in [5]. Where N candidate pairs are ranked by score in a task of m matching pairs it is defined as $\frac{1}{m}\sum_{i=1}^{n}\frac{c(i)d(i)}{i}$, where $c(i)$ is the number of correct pairs before rank position i and $d(i)$ is equal to 1 if the actual pair is a match, or 0 otherwise.

7 Experimental Results

7.1 Implementation

The method, both in its trained and nontrained forms was implemented in Java. Source codes are available at: privatewww.essex.ac.uk/~jhcama/TagLink.htm. We also implemented the tokenizer provided in the SecondString package.

7.2 The Data

Experiments were performed on each of the datasets listed in Table 4 and 3 other datasets randomly generated by the UIS database generator, [2]. The UIS database generator creates sets of records which are randomly corrupted. The level of random corruptions per record, the total number of records to be generated and the number of redundant records to be included in the artificial database are preset.

Table 4. Experimental data, source [5]

Dataset	Records	Redundancies
BirdKunkel	337	38
BirdScott2	719	310
Census	841	671
Cora	923	902
Parks	654	505
Restaurant	863	228

Table 5. Experimental results. Non-interpolated average precision of proposed methods VS best 14 methods. The best methods for each dataset are marked with "*". UIS column is the average result of the 3 randomly generated datasets

String metric	BirdKunkel	BirdScott	Census	Cora	Parks	Restaurant	UIS
Suggested	0.939	0.977	0.447	0.908	0.937	0.939	0.956
Suggested trained	0.955*	0.985*	0.469	0.920*	0.980*	0.980*	0.992*
SoftTFIDF	0.526	0.936	0.410	0.911	0.937	0.963	0.956
TFIDF	0.740	0.970	0.107	0.911	0.922	0.964	0.927
S.W.Gotoh	0.902	0.735	0.354	0.873	0.914	0.645	0.944
UnsmoothedJS	0.808	0.969	0.117	0.865	0.833	0.787	0.927
JelinekMercerJS	0.800	0.968	0.122	0.848	0.816	0.763	0.926
Jaccard	0.691	0.953	0.117	0.876	0.825	0.804	0.928
SmithWaterman	0.903	0.564	0.371	0.871	0.913	0.598	0.943
DirichletJS	0.608	0.965	0.121	0.861	0.832	0.765	0.926
MongeElkan	0.910	0.766	0.263	0.784	0.906	0.471	0.920
OverlapCoefficient	0.530	0.959	0.107	0.764	0.780	0.834	0.856
QGramsDistance	0.020	0.786	0.325	0.869	0.902	0.693	0.952
Level2JaroWinkler	0.055	0.531	0.484*	0.783	0.873	0.687	0.907
DiceSimilarity	0.165	0.729	0.112	0.791	0.772	0.754	0.861
CharJaccard	0.016	0.499	0.377	0.628	0.887	0.630	0.878

7.3 Experimental Methodology

Having N candidate pairs ranked by score in a task of m matching pairs, we measure the *non-interpolated average precision* as mentioned in section 6. We also measure the *maximum F1*, as $\max_i F1(i)$, [5], where $F1(i) = \frac{2*p(i)*r(i)}{p(i)+r(i)}$ is the harmonic mean at rank position i, $p(i) = \frac{c(i)}{i}$ is called the precision at position i, $r(i) = \frac{c(i)}{m}$ is called the recall at position i, and $c(i)$ is the number of correct pairs before rank position i.

We compare our non-trained method against methods contained in the SecondString and Simmetrics packages. Since the number of similarity methods to be evaluated is large, and since the number of all possible string pairs is $O(l^2)$, where l is the size of the dataset, the computational time required for the evaluation can be excessive. In order to reduce it, we compute a similarity score by a cheap string metric for each string pair in the dataset. If the score is greater than a certain threshold, then the string pair is kept for further testing, otherwise it

Table 6. Average precision and average maximum F1 of proposed method VS 14 selected methods. Sources: 1=SecondString, 2=SimMetrics

String metric	Precision	F1	Time (min.)	Source
Suggested	0.872	0.887	5.771	-
Suggested trained	0.897	0.895	8.334	-
SoftTFIDF	0.806	0.811	6.408	1
TFIDF	0.791	0.788	1.496	1
SmithWatermanGotoh	0.767	0.789	330.072	2
UnsmoothedJS	0.758	0.777	1.361	1
JelinekMercerJS	0.749	0.773	1.393	1
Jaccard	0.742	0.757	1.306	1
SmithWaterman	0.738	0.769	8.890	2
DirichletJS	0.725	0.739	1.391	1
MongeElkan	0.717	0.735	60.268	1
OverlapCoefficient	0.690	0.709	1.264	2
QGramsDistance	0.649	0.664	32.720	2
Level2JaroWinkler	0.617	0.658	8.069	1
DiceSimilarity	0.598	0.644	1.274	2
CharJaccard	0.559	0.580	1.855	1

is dropped. In our experiments, we use *cosine* [22] similarity as the cheap metric and 0.2 as its threshold. Recall that we consider each row of a dataset as an input string.

The domain dependent method is trained over both positive and negative training samples. We define the training set as stated in [18]. Positive training samples include all the real matching strings. Negative examples are selected by the non-trained method so that the closest estimated match are included in the training set, i.e. the non-match pairs with highest score. We sample a total of negative samples five times the positive sample size. As in [10] and [18], we split the available data into two, half for training and the other half for testing, and repeat the process with the sets interchanged.

Since the positive training set might be very small in some cases, the candidate list of matching characters might exclude some matches. To avoid this, we sample all possible different pairs of tokens in the dataset and obtain a matching score $s_{ij}(T_i, U_j)$, so that all matched pairs greater than a certain threshold Φ are included in the training set or as input for the parameter estimator algorithm. Here we set $\Phi = 0.7$.

7.4 Results

We report the evaluated precision of the 14 methods that performed best on average on all datasets. As shown in Table 5, the non-trained method performs best in 4 out of 7 cases, and its performance is very close to the best method in all other cases. The trained method performs best in all cases, except the census dataset, which is the most corrupted. The main advantage of the method proposed here is that it is consistent in its performance, unlike the other methods which show poor results in some cases. The average performance in both Precision and maximum F1 is shown in Table 6. In this case, both the trained and the non-trained methods perform in average better than the rest of the methods and the average computation time is also favorable compared to SoftTFIDF and Level2 hybrid methods.

8 Conclusions

We proposed a novel hybrid string metric, which selects matching pairs of tokens in a one-to-one setting, similar to the assignment problem. We believe this setting selects pairs of tokens in a better way than past approaches and it is computationally competitive. We use the same idea of assigning pairs of tokens to assign pairs of characters in order to define a new character based method. This method is combined with the Winkler scorer [23] and that improves its accuracy. For our token-based method, we define a variant of TFIDF weighting scheme which measures the ratio of matching characters common in pairs of strings. As mentioned before, this weighting scheme is better than TFIDF for the task of matching short strings.

The parameters of the proposed string metric can be estimated using the domain of the data to be processed. Although existing methods can perform very well in some cases, they can show a very poor performance in others. Our method, particularly when trained, performed consistently well, at least in all cases considered.

References

1. Newcombe, H.B., Axford, S., James, A.: Automatic linkage of vital records. Science **130** (1959) 954–959
2. Hernandez, M.A., Stolfo, S.J.: The merge/purge problem for large databases. In: SIGMOD: Proceedings of the International conference on Management of data. (1995)
3. Monge, A.E., Elkan, C.P.: An efficient domain-independent algorithm for detecting approximately duplicate database records. In: SIGMOD: Proceedings of the workshop on data mining and knowledge discovery. (1997)
4. Cohen, W., McAllester, D., Kautz, H.: Hardening soft information sources. In: KDD: Proceedings of the international conference on Knowledge discovery and data mining, Boston, Massachusetts, USA (2000)
5. Cohen, W., Ravikumar, P., Fienberg, S.E.: A comparison of string distance metrics for name-matching tasks. In: IJCAI and IIWEB, Acapulco, Mexico (2003)
6. Jaccard, P.: The distribution of the flora of the alpine zone. New Phytologist **11** (1912) 37–50
7. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM **18 issue 11** (1975) 613– 620
8. Jaro, M.A.: Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. Journal of the American Statistical Society **89** (1989) 414–420

9. Levenshtein, V.: Levenshtein distance algorithm. Keldysh Institute of Applied Mathematics, Moscow (1965)
10. Bilenko, M., Mooney, R.J.: Learning to combine trained distance metrics for duplicate detection in databases. Technical Report AI 02-296, University of Texas at Austin (2002)
11. Bilenko, M., Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. In: KDD: Proceedings of the international conference on Knowledge discovery and data mining, New York, NY, USA (2003)
12. Bilenko, M., Mooney, R.J.: Employing trainable string similarity metrics for information integration. In: IIWEB, Acapulco, Mexico (2003)
13. Bilenko, M., Mooney, R.J.: On evaluation and trainingset construction for duplicate detection. In: KDD: Proceedings of the Workshop on Data Cleaning, Record Linkage, and Object Consolidation, Washington DC, USA (2003)
14. Bilenko, M., Mooney, R.J.: Alignments and string similarity in information integration: A random field approach. In: Proceedings of the Dagstuhl Seminar on Machine Learning for the Semantic Web, Dagstuhl, Germany (2005)
15. Bilenko, M., Mooney, R.J., Cohen, W., Ravikumar, P., Fienberg, S.E.: Adaptive name-matching in information integration. IEEE Intelligent Systems **18 number 5** (2003) 16–23
16. Cohen, W., Richman, J.: Learning to match and cluster entity names. In: SIGIR: Workshop on Mathematical/Formal Methods in Information Retrieval, New Orleans, LA, USA (2001)
17. Leung, Y.W., Zhang, J.S., Xu, Z.B.: Optimal neural network algorithm for on-line string matching. IEEE Transactions on Systems, Man, and Cybernetics, Part B **28 number 5** (1998) 737–739
18. McCallum, A., Pereira, F.: A conditional random field for discriminatively-trained finite-state string edit distance. In: UAI: In Proceedings of the Conference on Uncertainty in Artificial Intelligence. (2005)
19. Monge, A.E.: An adaptive and efficient algorithm for detecting approximately duplicate database records. (2000)
20. Ristad, E.S., Yianilos, P.N.: Learning string-edit distance. IEEE Transactions on Pattern Analysis and Machine Intelligence **20 number 5** (1998) 522–532
21. Yancey, W.E.: An adaptive string comparator for record linkage. In: ASA: Proceedings of the Section on Survey Research Methods. (2003)
22. Chapman, S.: Simmetrics web intelligence, Natural Language Processing Group, Department of Computer Science, University of Sheffield, Regent Court, 211 Portobello Street, Sheffield, S1 4DP, UK. sam@dcs.shef.ac.uk. (2006)
23. Winkler, W.E.: String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In: ASA: Proceedings of the Survey Research Methods Section. (1990)

Image Processing

Applications

A Multi-Agent based Medical System with Several Learning and Reasoning Capabilities

Fernando Manzanares

Instituto Tecnologico de Cd. Madero-Universidad
Central de las Villas
fer_manzanares@yahoo.com

Abstract. I present a Multi-agent System that represents a Medical community where each Doctor may have a different specialty and may do his/her work with specific techniques. In this sense, I propose the use of several Machine Learning techniques as a form to represent the use of different techniques to learn to diagnose diseases and the use of several data sources to produce knowledge, permitting that each data source can be related with some specific medical specialty. Also I propose the separation of the processes of learning and reasoning to increase the use of Machine Learning techniques whose implementations requires great amount of computer resources; for this, the knowledge extraction from data is done by software agents that are executed in high performance computers while the reasoning processes that exploit the knowledge discovered are carried out by software agents that are executed in average scale computers.

1 Introduction

In each community we can find doctors with perhaps different specialties: ophthalmology, bacteriology, cardiology, among many others. Also, each doctor could have attended his studies in a different University and have made his specialization in a different Hospital than the others. As well, each one can set in an independent way the amount of its fees. Nevertheless, the great majority of works related to processes of knowledge extraction from data do not follow this behavior since they are focused in the implementation of a single technique and/or in the creation of a model associated with a particular problem domain. In this work an analogy to the situations that occur in the real life is created through the construction of a multi-agent based medical system, where the agents have multiple capabilities of learning and reasoning. In their construction several machine learning techniques are used such as Artificial Neural Networks, Rough Set Theory, Bayesian Learning and ID3 algorithm, as well as reasoning processes according to the models that the previous techniques build.

The purpose is simulating the different approaches used by each doctor to carry out its work. Also several decision systems related to different medical areas are used in

© A. Gelbukh, S. Torres, I. López (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 19, 2006, pp. 211-224

order to create different artificial specialists. As development platform was utilized Java programming language and JADE framework.

2 Machine Learning

2.1 Problems Related to Machine Learning

During the last decades an enormous increment in the use of sensors as well as in the development of data storage devices has been presented. This has produced an enormous amount of data, available to be analyzed by specialists to be able to understand and to control in a better way the underlying problem domains. On the other hand the increment in the power of available computation for these specialists has done possible the use of more complex models, causing among others things an increase in the investigation and development in diverse fields of the Artificial Intelligence [1].

Within the techniques used in Artificial Intelligence for problem domains understanding the machine learning is found, with an extensive use of supervised inductive learning, which is used to acquire knowledge from examples previously classified.

Many techniques of supervised inductive learning have been used, within that we can mention: Artificial Neural Networks, Rough Set Theory, Bayesian Learning, Genetic Algorithms, the family of algorithms derived from ID3, among others. Each one of them has its strengths and its weaknesses. For example, it is well known that decision trees produced by ID3 are highly understandable and expressive, nevertheless is also known the incapacity of ID3 to deal with unstable, uncertain or incomplete data. On the other hand the Artificial Neural Networks are excellent universal approximators with capacity to process incomplete or noise data, but the models generated by them as a form of knowledge representation in numerical matrices form makes no sense for a human being.

Another situation that we must consider as advantages and disadvantages of these techniques consists of the type of inputs that are able to process. For example, techniques such as ID3 or Bayesian Learning are not adequate to process images unlike the Artificial Neural Networks.

Another disadvantage of the majority of the Machine Learning techniques is the great amount of computational resources that they require for his execution. This diminishes the number of potential users that can be benefited from their use.

It is by that a multi-agent system in which some agents takes charge of the process of supervised inductive learning and other agents to use the models created by the first in order of putting in use the acquired knowledge. Creating with this an artificial medical community.

2.2 Supervised Inductive Learning

Many induction problems can be described as follows [2]. One begins with a training set of preclassified examples, where each example (also called observation or case) is described by a vector of values of characteristics or attributes, and the objective is to form a description that can be used to classify with high precision examples non previously seen. In a formal way we say that an example is a pair *(x, f(x))*, where *x* is the input and *f(x)* is the output of the function applied to *x*. The objective of the inductive inference is, given a set of examples of *f*, to produce a function *h* that approximates *f*. Normally *x* is a vector of attributes each one with a particular domain and function *f* is the valuation done by a human expert of the values of *x*. For example *x* can be a set of symptoms, vital signs and results of biochemical analysis of human patients and the output *f(x)* can be the diagnosis done by a doctor.

According to [9] the construction of a procedure of classification from a data set for which the classes are known has also been called in an indistinct way as pattern recognition, discrimination, discovery of knowledge or supervised learning. Being distinguished of the non supervised learning or grouping in which the classes are inferred from the data.

2.2.1 Decision Systems

Independently of the used technique to carry out the induction, the sets of examples that are used can be presented in a standard form of a Decision Table, which is an implementation of a Decision System.

Given an Information System, defined like a pair A = (U, A), where U is a non empty finite set of objects called universe of examples (objects, entities, situations or states, etc.) and A ={A1, A2,...,An} is a non empty finite set of attributes, such that the elements of U are described using the attributes Ai. If to each element of U a new attribute d called decision is added, indicating the decision taken in that state or situation, then a Decision System (U, A∪{d}, where d∉A) is obtained [2]. The values of the decision attribute d are, as already was mentioned, the outcome done by a human expert. For example, the Ai□A attributes can be the symptoms or characteristics of the patients whom go to medical consultation in a particular clinic and the values of d can be the diagnosis done by the doctors of the clinic whom attended each patient.

3 Artificial Neural Networks

Artificial Neural Networks (ANNs) are known by diverse names, among them: connectionist models or parallel distributed processing models. Instead of executing a program sequentially as in a Von Neumann architecture, the ANN explores many hypothesis simultaneously using massively parallel networks of many elements of processing connected by weighted connections. The ANNs are inspired in the biological model of the human brain, without reaching to duplicate it. The main purpose of all the Biological Neural Systems is the centralized control of several biological functions, some of them responsible for supplying of energy, therefore the neuronal system is connected with the metabolism, the cardiovascular control and the breathing. In the human beings, as well as in the majority of the superior animals, the greater capacity of the neurological system is related to the behavior, this is, the control of the state of the organism with respect to its environment [4, 5].

In the cerebral nervous system the central element is a cell called neuron. An important difference of these cells with respect to the rest of the alive cells is its capacity to communicate. In general terms a neuron receives input signals, combines and integrates them and emits a output signal. In fact a single neuron generally does not do anything, performance and results are determined by the cooperative work of several of these neurons.

The ANNs normally consist of Artificial Neurons as the one that is shown in figure 1. The Artificial Neuron is seen like a node connected with other by means of links that correspond to connections axon-synapse-dendrite, which are present in the biological neuron.

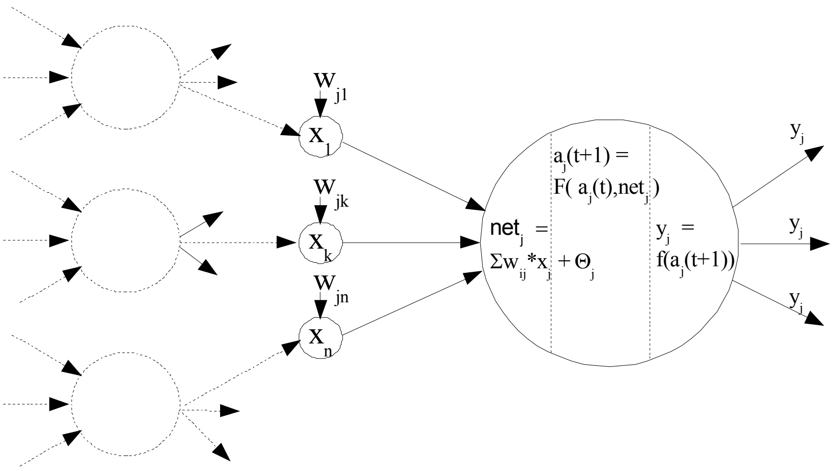


Fig. 1. Artificial Neuron

There is associated a weight with each connection. As in the case of the synapse in the biological neurons that weight determines the nature and intensity of the influence of one node on another one. In more specific way, the influence of one node on another one is the product of the signal of the input neurons by the weight of the connection that connects them with the node in which influences. For example, a large positive weight corresponds to a strong excitation, and a small negative weight corresponds to a weak inhibition. This interaction causes that in every single moment of time t all the neurons that compose the ANN be found in a certain state. In a simplified way, we can say that there are two possible states: rest and excitation, that we will call activation states. These activation values can be continuous or discreet. In addition, they can be limited or unlimited. If, for example, they are discreet binary, an active state would be indicated with a 1, and is characterized by the emission of an impulse by the neuron (action potential), while a passive state would be indicated for a 0 and would means that the neuron is in rest.

To generate the state of activation aj(t) of each node j, these combine the individual influences that receive in their input connections in a single global influence, by means of an activation function. A single activation function passes the weighted sum of the input values through a transfer function to determine the output of the node. In case of production of binary outputs, this can be 0 or 1, depending on if the weighted sum of inputs is down or above the threshold value utilized by the node's activation function.

The connections that link the neurons that form the ANN have an associated weight, which is where the knowledge acquired by the network is placed. We consider the case where a neuron j is connected in its inputs with N units. Let us denominate wji the weight on the connection between the neuron i and the neuron j. Also we denominate xi the output value that neuron i transmits to neuron j. As simplification we consider that the effect of each input signal is additive, in such way that the net input netj that receives a neuron is the sum of the product of each individual signal by the value of the synapse that connects both neurons:

$$\mathrm{net}_{\mathrm{j}} = \sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{w}_{\mathrm{ji}} \cdot x_i - \Theta_i$$

This rule determines the general procedure to combine the input values of a unit with the weight of the connections that arrive at the same one and is known as propagation rule, where □j represents the threshold value of the neuron or a bias term on it. Also a called activation rule exists [7], the one that determines how combines the value of the weighted inputs netj and the present state of the neuron a(t – 1) to produce a new state of activation:

$$\mathrm{a_j(t) = F(a_j(t-1), net_j)}$$

This function F produces a new state of activation in the neuron j from the state in the previous instant t-1 and the combination of the weighted inputs in the present instant t.

In most cases F is the identity function, reason by which the state of activation of a neuron will be the value netj of the same one. In this case, the parameter that is passed to the output function will be netj, directly, without being taken into account the previous activation value. In agreement with this the output yj of a neuron j will be according to the expression:

$$\mathrm{y_j(t) = f(net_j - \Theta_j) = f(\sum_{i=1}^{N} w_{ji}} y_i(t-1))$$

The modification of the weights of a network can be carried out in diverse ways, grouped in two great classes: supervised learning and unsupervised learning. Within the main types of supervised learning we have:

a) Learning by correction of errors. It consists of adjusting the weights in function of the difference among the output values from the network and the expected values. One of the error correction rules more extensively used is the called Generalized Delta Rule or Gradient Descent Rule, base of Backpropagation algorithm:

$$\Delta w_{ij}(t+1) = \alpha(d_{pj} - y_{pj}) f'(net_j - \Theta_j) y_{pi}$$

where
Δw_{ij}: is the amount by witch to change the weight of the connection between neurons i and j

α: is the learning rate parameter
p: is the pth training example
d_j: is the desired output of neuron j
y_j: is the output of neuron j
Θ_j: is the threshold value of neuron j

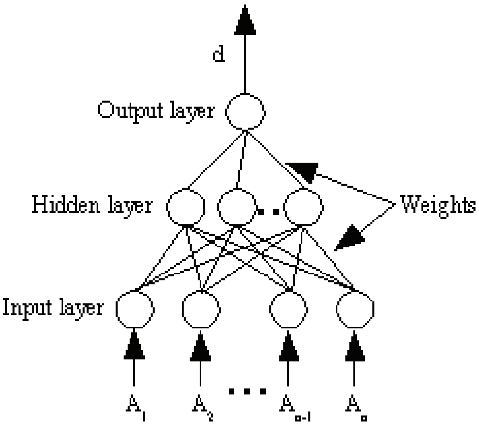


Fig. 2. A Multi-Layer Perceptron

b) Learning by reinforcement. It is a subtype of supervised learning, characterized by there are not complete examples of the desired behavior, but a supervisor exists that simply informs by means of a reinforcement if the network output is adjusted or not o the desired output and, in function of it, the weights used are adjusted using a probabilistic base.

c) Stochastic learning. This type of learning consists of making random changes to the weights values, evaluating its effect from the desired objective and based on probability distributions [8].

Attending its relation with the environment, there are four types of artificial neurons: input, output, hidden and composite. Input neurons are which receive from the external environment the information that the ANN will learn or will process. Output neurons are the units that are responsible to carry out the results of the processing done by the ANN in the input data. Hidden neurons do not have any contact with the external environment of the ANN. Composite neurons are input/output neurons.

We say that the neurons of the same type are grouped in a layer. Therefore, the ANNs could have one or more layers forming the network topology. Besides, related with the form in which the signals are transmitted over the network, there are feed-forward, feedback and recurrent connections.

The kind of ANN used in this work is the Multi-Layer Perceptron (MLP) using the backpropagation algorithm. This is a multi-layer network, with feed-forward connections, with continues inputs and outputs lessen in the interval [0, 1], and which uses the Generalized Delta Rule as learning rule. Here the MLPs used are restricted to only one hidden layer and one output. This is illustrated in figure 2.

3.1 Rough Set Theory

Rough Set Theory was introduced by Zdzislaw Pawlak in 1982 [10] as a formal mathematical theory for the modeling of the knowledge about a domain of interest in terms of a collection of equivalence relations. Its main area of application is in the acquisition, analysis and optimization of computer processable models from data. The models can represent functional, partial functional ant probabilistic relations existing in data through the extended approaches of the Rough Sets.

This theory often has been proved to be an excellent mathematical tool for the analysis of vague descriptions of objects, particularly when the vagueness refers to inconsistencies or ambiguities due to the level of information granularity [12]. Its importance in the Artificial Intelligence and the Cognitive Sciences is related to the areas of machine learning, knowledge acquisition, decision analysis, knowledge discovery from databases, expert systems, decision support systems, inductive reasoning and pattern recognition [5].

The starting point of the philosophy of the rough sets is the assumption that with each object of interest, in a problem domain, exists some associated information. For example, if the objects are patients that suffer a particular disease, the symptoms of the patients form the information about the patients [10].

In [11] it is mentioned that the problems that can be resolved with the rough set theory are:

1. characterization of a set of objects in terms of attributes values
2. total or partial dependencies, among attributes
3. reduction of attributes
4. significance of attributes
5. decision rules generation

One of the main advantages of this theory is that it does not require to having any preliminary or additional information about the data, just as probability distributions in statistics, assignment of basic probabilities in the Dempster-Shafer theory or membership functions in the fuzzy set theory.

From the contained information in the decision system, is desired to discover rules of the form X □ Y (C), where X is the condition or antecedent of the rule built from the attributes of the data set A, Y is one of the values that can take the decision attribute d, and C is the certainty of the rule.

From an equivalence relation B and the use of upper and lower approximations (rough sets) such rules can be constructed. Any subset of A (B □ A), can be used as equivalence relation, although the use of the attributes of A with greater relevance or importance in the application domain are recommended. The idea is to construct the Yi sets, in which are all the elements of U which have same value yi of the decision attribute. To these sets Yi their upper and lower approximations are determined and from them the rules are generated. The algorithm of construction of the rules is the following one (LRRS, Learning Rules using Rough Set) :

1. Build the decision system $(U, A \cup \{d\})$.

2. Define the subset $B \subseteq A$ of attributes that are considered relevant.

3. Build the sets $Y_j \subseteq U$, such that in Y_j are all the elements of U that have y_i as value in the decision attribute.

4. Build the equivalence classes X_i from the relation B.

5. Build the lower and upper approximations for each subset $\mathrm{Y_j}$:

$$B_*(Y_j)= \{x \in U / B(x) \subseteq Y_j\}$$
$$B^*(Y_j)= \{x \in U / B(x) \cap Y_j \neq \}\phi$$

6. Build the limit region of each subset Y_j for the equivalence relation of B:

$$BNB\ (Yj) = B^*\ (Yj) - B^*\ (Yj)$$

7. Build rules of certainty 1.

```
For all $X_i$ do
    For all $Y_i$ do
        If $X_i \subseteq B_*(Y_j)$ then generate the rule: $X_i \Rightarrow Y_j\ (1)$.
```

8. Build rules of certainty smaller than 1.

```
For all $X_i$ do
    For all $Y_i$ do
    If $X_i \subseteq BN_B(Y_j)$ then generate the rule: $X_i \Rightarrow Y_j\ (C)$,
    where $C=/X_i \cap Y_j//X_i$
```

The previous algorithm presents a simple form to build rules. Nevertheless, from the use of rough sets more complex procedures to knowledge discovery have been developed.

3.2 Bayesian Learning

A Bayesian Classifier is trained by the estimation of the conditional probability distribution of each attribute, producing the label of the class, from the database. A case is classified from its set of attributes values, using the Bayes' rule. The case then is placed in the class with the greater probability. The underlying assumption that simplifies the Bayesian classifiers is that the classes are exhaustive and mutually exclusive and that the attributes are conditionally independent once the class is known. A Bayesian classifier is defined by a set C of classes and by a set A of attributes. We denote a generic class as c_j, and a generic attribute as A_i. The set C of classes can be tried as a stochastic variable taking one of the values c_i with a probability distribution that represents a unknown state of the world. The decision system is used to determine the probabilities $P\ (c_j)$ and $P\ (A_i|c_j)$ for each attribute A_i. These probabilities are determined counting the number of instances. All the attributes values depend on their class only, and connections between attributes are not permitted[3]. Figure 3 shows a Bayesian Network that is used like a Bayesian classifier.

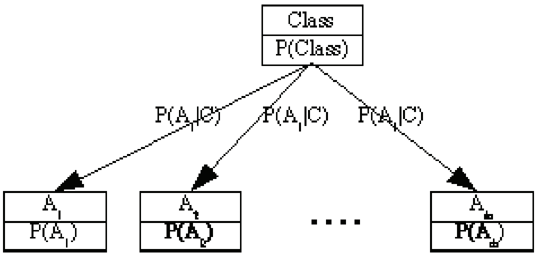


Fig. 3. Bayesian network.

Related with each link in the Network exists a Conditional Probability Table such that is shown in table 1. Suppose that you observe a new case with A1 = v2, A2 = v2,..., Ak = vk. We use the Bayes' rule to determine the posterior probability of the class cj of the new case, conditioned in the attributes values as follows:

$$P(c_j \mid A_1 = v_1, A_2 = v_2, ..., A_k = v_k) = P(A_1 = v_1, A_2 = v_2, ..., A_k = v_k \mid c_j) P(c_j) / P(A_1 = v_1, A_2 = v_2, ..., A_k = v_k)$$

Using the independence assumption this is simplified to:

$$P(c_j \mid A_1 = v_1, A_2 = v_2, ..., A_k = v_k) = P(A_1 = v_1 \mid c_j) * ... * P(A_k = v_k \mid c_j) / P(A_1 = v_1, A_2 = v_2, ..., A_k = v_k)$$

The values P(A1 = v1 | cj) are obtained from the conditional probability tables. The denominator P(A1 = v2, A2 = v2, ..., Ak = vk)

is a normalization factor to force the addition of probabilities is one.

Table 1. Conditional Probability Table

$P(A_1 \mid C)$	c_1	...	c_n
$a_{1,1}$	$P(A = a_{1,1} \mid c_1)$	...	$P(A = a_{1,1} \mid c_n)$
$a_{1,m}$	$P(A = a_{1,m} \mid c_1)$		$P(A = a_{1,m} \mid c_1)$

3.3 ID3 Algorithm

This algorithm produces knowledge in a decision tree representation. Learning of trees is a method to approximate functions of discrete values. A decision tree classifies the instances ordering them top-down form root to leaves. Each internal node of the tree specifies a test from an attribute and leaves are the classes in which instances are classified. Each link of an internal node corresponds to some possible value of the attribute tested in that node. A decision tree represents a disjunction of conjunctions on the attributes values. Each branch from the root to a leaf node corresponds to an attribute conjunction, and the tree is itself a disjunction of that conjunctions. This is illustrated in figure 4.

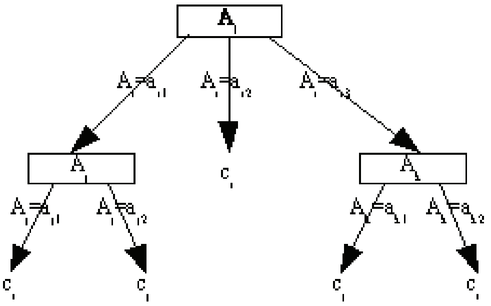


Fig. 4. A decision tree

The ID3 is a recursive algorithm:

```
ID3(DS:Decision System) : Decision tree
    1. Build a root node.
    2. If all examples have the same decision $d = d_i$ then
    - Label root node with $d_i$
    - Return root node
    3. If attribute list A is empty then
    - Label root node with most common value of d
    - Return root node
    4. bestNode ← attribute with minimum entropy
    5. Label root node with bestNode
    6. For each value $v_i$ of bestNode do
    6.1 add new descendant branch to bestNode related with test bestNode = $v_i$
    6.2 Let $examples_{vi} = \{X \mid X \in U \wedge bestNode(X) = v_i\}$
    If $examples_{vi}$ is empty then
    Build a leaf node
    Label the leaf node with the most common value of d in U
    Link the descendant branch with the leaf node
    else Build a new node = ID3({$examples_{vi}$,
    A-bestNode, {d}})
    Link the descendant branch with new node
    7. Return root node
```

4 Architecture of the multi-agent based medical system

4.1 System Architecture

The Multi-agent System proposed consists of the development, in first place, of at least four agents with inductive machine learning capacity, based on each one of the techniques mentioned in the section 2 of this document, with the additional capacity to deliver the models created to other agents who therefore requested it. These agents should be executed in a high scale computer that includes a Java virtual machine based on the J2SE and the JADE framework.

In second place the development of at least other four agents with capacity of reasoning is proposed. This type of agents should interact with the learning agents to request the models that can be employed in the solution of specific problems, for which these must know what models are available for be able to reason with each one of them. Besides these agents can be consulted for some another agent that have not capacity of reasoning and whose has the roll of avatar of a human being. Reasoning agents also can be avatars, and must have graphical user interfaces. Reasoning agents can be executed in average scale computers that includes the J2SE Java Virtual Machine and the JADE framework.

The third type of agent that is proposed only has the capacity to consult models available, verify if those are adequate to particular problem, and then acquire the data

related with the problem instance to be sent to a reasoning agent for its solution. It is for this that them receive the User Interface Agent or Avatar Agent name. These should be capable of interacting as with learning agents to know the available models, as with reasoning agents to request them reasoning services. For their execution low scale computers can be used, such as PDAs or Smart Phones, which include the J2SE or J2ME Java Virtual Machine and the JADE framework. Figure 5 shows this plan.

4.2 System Operation

The process initiates when learning agents carry out the process of induction on a subset of learning of a decision system related to cases of some medical specialty. Once it extracted the knowledge this is validated on the base of a subset of test of the same decision system. A human expert must provides to each learning agent the goal to obtain a minimum percentage of effectiveness. The minimum effectiveness can be established as a set:

effectiveness = (minimum effectiveness rate, minimum false positives rate, minimum false negatives rate)

Once the agent has reached the goal, should publish in the yellow pages service the technique used, the price of its services and the name of the specialty related to the decision system from the knowledge was extracted. The price is used as comparison base for selection in the event that two or more agents have produced models related to the same specialty. In this work, the price is calculated in terms of the effectiveness, for it the same expert that established the minimum effectiveness should establish how affect false negatives and false positives in the associated specialty, therefore the price published is: Price = AccRate - w1 * PosRate - w2 * NegRate

where

AccRate : accuracy rate,
PosRate : false positives rate,
NegRate : false negatives rate,
w_1 *: weight associated with PosRate,*
w_2 *: weight associated with NegRate*

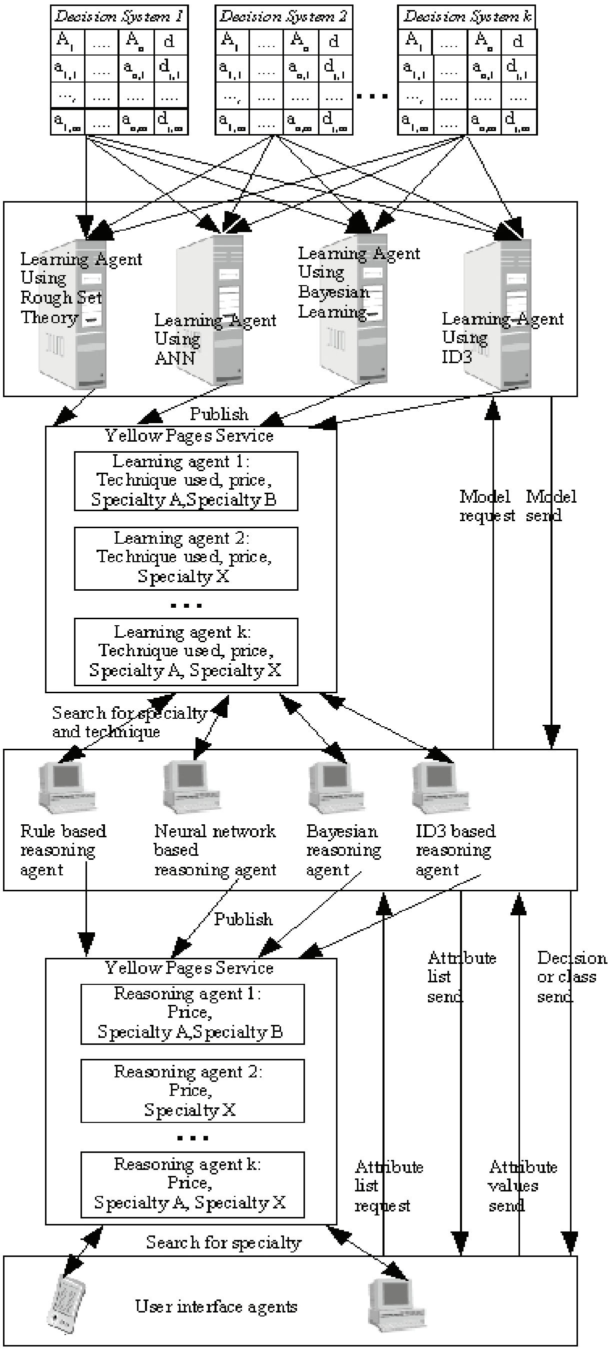

Decision System 1
Decision System 2
Decision System k
A1 .... An d
a1,1 .... an,1 d1,1
...
a1,m .... an,m d1,m
Learning Agent Using Rough Set Theory
Learning Agent Using ANN
Learning Agent Using Bayesian Learning
Learning Agent Using ID3
Publish
Yellow Pages Service
Learning agent 1: Technique used, price, Specialty A,Specialty B
Learning agent 2: Technique used, price, Specialty X
Learning agent k: Technique used, price, Specialty A, Specialty X
Model request
Model send
Search for specialty and technique
Rule based reasoning agent
Neural network based reasoning agent
Bayesian reasoning agent
ID3 based reasoning agent
Publish
Yellow Pages Service
Reasoning agent 1: Price, Specialty A,Specialty B
Reasoning agent 2: Price, Specialty X
Reasoning agent k: Price, Specialty A, Specialty X
Attribute list send
Decision or class send
Attribute list request
Attribute values send
Search for specialty
User interface agents

Fig. 5. System architecture

As each technique produces a particular knowledge representation and since the operation of knowledge extraction will do it another agent, then the learning agent must publish the used technique so that only the agents who use the knowledge representation associated with the corresponding technique can offer the medical consult service related to the underlying specialty.

Therefore, each reasoning agent is capable of using a reasoning method related with a knowledge representation. Each reasoning agent that wants to classify a new case will request the corresponding model created by the learning agent. Each model will be sent codified according to the type of representation created by the learning agent. In this case four types exist, as already we know:

1. Network weights. In this case the learning agent delivers to the reasoning agent the network weights in form of two numerical matrices. Preceding the matrices, the data of the network structure is sended: number of inputs and number of hidden units. The reasoning process consists of using the weights matrices to calculate the network output. To be able to do this, is also necessary that the learning agent send the attributes domain both inputs and decision, as well as the form to codify the input attributes and to decode the network output.
2. Rules with certainty factors obtained with the algorithm LRRS. In this case the set of rules is sent to the reasoning agent that serves as knowledge base. Each rule is of type:
IF $A_1 = a_1$ *AND ... AND* $A_n = a_n$ *THEN* $d = d_i$ *(C)*
3. Bayesian network. In this case the learning agent sends to the reasoning agent the set of attributes A, the domain of each attribute A_i, the domain of the decision attribute d and the tables of conditional probability for each combination attribute-class.
4. Decision tree. In the case of the models produced by ID3 the decision tree generated is sent to the corresponding reasoning agents.

In cases 2 and 4, the set A of attributes as their domains also must be sended to the reasoning agents.

Finally, reasoning agents can play the roll of avatar of a human being, but another agent with this particular roll must be developed, specially if we want use a low scale computer, such as a PDA or a cell phone.

JADE framework provides all the functionality needed for agents creation, communication between agents and yellow pages service; among other functionalities related with agent oriented develop.

4 Conclusions

This approach was developed mainly with the purpose of creation of an artificial medical community in which may be possible the implementation of multiple machine learning approaches. Tests done have demonstrated that the effectiveness rate of the four techniques used are very similar in the training phase, except the ANNs which performance has depended of correct selection of its characteristics parameters.

However, in the operation phase the performance vary significantly among them. The best situated have been the Bayesian Classifier, mainly when not all the attributes values are available. The other three models have had similar performances.

The distributed approach has permitted the use of machine learning models in middle and low scale computers, increasing their use.

We will have to do more work in the future, in order to incorporating more machine learning techniques, as well as another approaches of creating artificial communities.

References

1. Baldi, P. y Brunak, S. 2001. *Bioinformatics: the Machine Learning Approach.* The MIT Press. Cambridge, Massachusetts, 2001.
2. Bello, R. y Gaitán, J.J. *Tomando Decisiones Basadas en el Conocimiento.* Universidad Cooperativa de Colombia, 2003.
3. Cantú, F. J. *Learning and Using Bayesian Networks for Diagnosis and User Profiling.* CIA-ITESM, 2000.
4. Freeman, J. A. y Skapura, D. M. *Neural Networks. Algorithms, Applications and Programming Techniques.* Addison-Wesley Publishing Company, Inc., 1991.
5. Grzymala-Busse, J. W. And Ziarko, W. Data Mining Based on Rough Sets. In *Data Mining: Opportunities and Challenges.* Idea Group Publishing, 1994.
6. Jordan, M. I. and Bishop C. M. *Neural Networks.* CRC Handbook of Computer Science, CRC Press. Boca Raton, FL., 1996.
7. Kröse, B. and Van Der Smagt, P. *An Introduction to Neural Networks.*The University of Amsterdam., 1996.
8. Masters, T. *Advanced Algorithms for Neural Networks. A C++ Sourcebook*. John Wiley & Sons, Inc. Canada., 1995.
9. Michie, Spiegelhalter, D.D.J. and Taylor C.C. *Machine Learning, Neural and Statistical Classification.* MRC Biostatistics Unit, Institute of Public Health, University Forvie Site, Robinson Way, Cambridge CB2 2SR, U.K., 1994.
10. Pawlak, Z., Grzymala-Busse, J. Slowinski, R. and Ziarko, W. *Rough Sets.* Communications of the ACM. November 1995. Vol. 38, N° 11., 1995.
11. Pawlak, Z. *Rough Sets and Data Analisys.* Proceedings of the 1996 IEEE Fuzzy Systems Simposium, 1996.
12. Voges, K. E., Pope, N. K. Ll. and Brown, M. R. Cluster Analysis of Marketing Examining On-line Shopping Orientation: A Comparision of k-means and Rough Clustering Approaches. In *Heuristic and Optimization for Knowledge Discovery.* Idea Group Publishing, 2002.

A CBR Diagnostics System
Applied in the Brazilian Public Health System

Márcia Regina Moss Júlio[1], Gilberto Nakamiti[2]

[1]R. Clariano Peixoto, 280 – Limeira – SP - BRAZIL
Communitary Faculty of Limeira
lmjulio@linkway.com.br
[2]Rod. D. Pedro I, Km 136– Campinas – SP - BRAZIL
Catholic University of Campinas / Paulista University
g_nakamiti@uol.com.br

Abstract. Case-based systems constitute an Artificial Intelligence technique, which is well-known for their ability to reuse and to adapt past experiences to solve new problems. To achieve this, they store and retrieve past experiences, based on their attributes, adapt them, and apply the adapted solution to the new problem. After analysing the solution performance, it may store the new case in the case-based, so that the system can improve its performance over time. The main feature of such systems is the retrieving mechanism, which is responsible for comparing and identifying similar, relevant information in the stored cases, through similarity metrics. This work presents an approach for handling multivalued attributes in well-known similarity metrics in order to retrieve the most relevant cases for a specific class of application. A health area application concerning lateral epicondilitis, which is an elbow tendonitis, was developed, and it illustrates its use. This work contains real data from the Brazilian public health system, where the work was developed and is been prepared to be used.

1 Introduction

When we face a new problem, we often remind similar problems that we solved in the past, and use them to solve the new one.

Case-Based Reasoning (CBR) may be seen as a problem solving approach that uses and adapts similar past solutions for new problem situations.

The mechanism used to retrieve the best and most similar past situations is of utmost relevance in such systems, because the situations retrieved will be used as the basis for the new problem solution.

This work presents an approach to handle multivalued attributes in well-known similarity metrics, and it shows its use in a health area case-based system,. This application had a strong demand on the Brazilian public health system, in particular in Araras and Limeira regions, in the state of Sao Paulo, Brazil, due to the lack of personnel in some public health facilities.

Section 2 presents an overview of the main case-based concepts and similarity metrics used in this work.

Section 3 describes the application, and shows how the multivalued attributes were handled, besides the results obtained by the system.

Finally, we present the conclusions of this work.

2 Case-Based Systems

Case-Based Systems are an Artificial Intelligence (AI) technique that reproduces human cognition aspects to solve specific problems that usually are solved by human experts in a specific knowledge area. They simulate the human act of remembering a past case to solve a new one through the identification of affinities among them [2] [6] [7].

In a Case-Based System, a case represents a complete description of a problem in that application domain, with a solution already applied. It is an abstraction of an experience, described by valued attributes. The attributes describe the experience content and context. After a similar case is retrieved based on its attributes, it is adapted to fit the new problem, offering a better solution [3] [12] [13].

The CBR cycle proposed by Aamondt e Plaza, illustrated by Figure 1, is very useful to understand the process, being composed by four tasks:

- Retrieve the most similar cases;
- Reuse (adapt) the solution for the new problem;
- Revise the proposed solution; and
- Retain (learn) the experience representing the new case for further use.

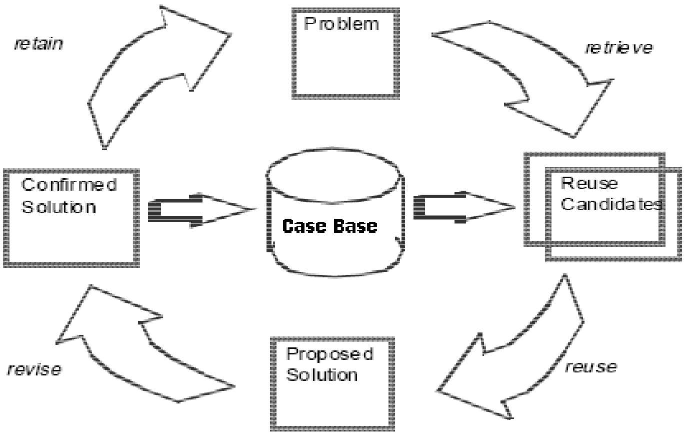


Fig. 1 The CBR Cycle

Similarity metrics are of utmost importance in case-based systems because they are responsible to identify the most similar stored solutions to be used in the new problem situation. These solutions will serve as basis for the whole cycle [10] [11].

Similarity may be seen as an intuitive concept used to describe the perception of a human observer about the common aspects existing in two objects [14].

A very accepted formalization of the similarity concept in computer systems resides in the definition of a numeric measure of distance or similarity. We can understand the similarity measure as a formalization of a specific similarity judgment through a concrete mathematical model [4].

A similarity metric synthesizes the similarity through a measure of importance of each attribute when compared with the other ones, and this process can be conducted in a variety of ways [5] [8] [9].

3 The FisioSmart System

In order to develop the application system, we first conducted extensive interviews with health area professionals from public health facilities in Araras and Limeira regions, in the state of Sao Pauloi, Brazil, to better understand the application domain. These professionals needed an automated system to help their work and of their assistants to take care of the poor people of those regions. The first focus was to identify the relevant cases attributes, and their correlations among themselves and the possible diagnoses.

3.1 Application

FisioSmart is a Diagnostics Evaluation System for Lateral Epicondilitis, which is an elbow tendonitis. The application developed uses CBR and serves as a basis to test the efficacy of similarity metrics, in particular the ones based on distance handling multivalued attributes.

The system was developed in Delphi, using a Firebird Database.

When a physiotherapist interviews a patient with lateral epicondilitis, he or she formulates a series of questions to aid the choice of the best procedures and conducts to treat the patient. With the patient answers, the system will be able to retrieve the most similar cases in the case-base, in order to provide a good similar case to be reused.

3.2 Implemented Metrics

The similarity metrics implemented in FisioSmart were:

- City-Block (Manhattan);
- Characteristics Count;
- Euclidian;

- Square Euclidian;
- Weighted Square Euclidian;
- Closest Neighbor – Query Insert Function;
- Closest Neighbor – Intersection Function;
- Closest Neighbor – Linear Function.

When we applied some of the cited similarity metrics, in particular when dealing with multivalued attributes, some adjustments were performed, as they will be described in the next sections.

3.3 Dealing with Monovalued and Multivalued Attributes

In order to test and to compare the similarity metrics implemented, we started from 40 real cases, taken from public health centers in Araras region, state of Sao Paulo, Brazil.

From the 40 cases collected, 25 cases fed the case-base, and 15 cases were used as new cases. The system had to propose treatments for these new cases. Each test set consisted of inserting the 25 cases in the case-base, and simulating the 15 new ones. This test set was performed for each of the similarity metrics implemented.

At this point, some considerations should be made:

1. Some monovalued attributed were classified as Boolean, and the Boolean metrics was used to handle them, independently of which metrics was been used with the other attributed [10]

2. Each multivalued attribute could use a set of predefined values.

Moreover, to apply the similarity metrics some adjustments in the original metrics were performed:

1. Usually, the similarity metrics use the descriptors distance to calculate the attributes similarity value. This strategy is very useful when handling monovalued attributes, but it is not usable in the original form to handle multivalued attributes. The first proposed adaptation consists in analyzing the set of valued, as presented in Table 1.

Table 1. Adjustment 1

	New Case		**Case in the Case-Base**	
	Attribute – Main Symptom	**Descriptor value**	**Attribute – Main Symptom**	**Descriptor value**
01	Elbow pain	0.7	Elbow pain	0.7
02	Itching	0.8	Itching	0.8
			Fist pain	0.6

To compare the multivalued attribute Main Symptom, we used a Cartesian product among the stored attributes' values and the new cases ones because, for this specific application area, the symptoms are not independent, as they reflect in one another. Due to this interdependency, they were rated based on their mutual proximity, so that the closer the symptoms, the closer their values. We can observe below a model of how this strategy is conducted:

1-(L1N – L1B) → 1- |0.7 – 0.7| = 1
1-(L1N – L2B) → 1- |0.7 – 0.8| = 0.9
1-(L1N – L3B) → 1- |0.7 – 0.6| = 0.9
1-(L2N – L1B) → 1- |0.8 – 0.7| = 0.9
1-(L2N – L2B) → 1-|0.8 – 0.8| = 1
1-(L2N – L3B) → 1-|0.8 – 0.6| = 0.8

Average → (1+0.9+0.9+0.9+1+0.8)/6 = 0.91
Maximum → 1

Where, in Table 01:

L1N → Row 01 of the New Case
L2N → Row 02 of the New Case
L1B → Row 01 of the Case in the Case-Base
L2B → Row 02 of the Case in the Case-Base
L3B → Row 03 of the Case in the Case-Base

2. When comparing two multivalued attributes with identical values, the Cartesian product may lead to a misleading result. Table 2 shows an example.

Table 2. Adjustment 2

	New Case		Case in the Case-Base	
	Attribute – Main Symptom	**Descriptor value**	**Attribute – Main Symptom**	**Descriptor value**
01	Elbow pain	0.7	Elbow pain	0.7
02	Itching	0.8	Itching	0.8

Since the two attributes are identical, the similarity measure between them should be 1. This simple adjustment only turns to 1 the similarity value measured between identical multivalued attributes.

1-(L1N – L1B) → 1- |0.7 – 0.7| = 1
1-(L1N – L2B) → 1- |0.7 – 0.8| = 0.9
1-(L2N – L1B) → 1- |0.8 – 0.7| = 0.9
1-(L2N – L2B) → 1- |0.8 – 0.8| = 1

Average → (1+0.9+0.9+1)/4 = 0.95
Maximum → 1

Where, in Table 2:

L1N → Row 01 of the New Case
L2N → Row 02 of the New Case
L1B → Row 01 of the Case in the Case-Base
L2B → Row 02 of the Case in the Case-Base

Table 3 shows an example of how the global case similarity function is calculated in a case example, with monovalued, multivalued and Boolean attributes:

Table 3. Calculating the similarity of a case

Attribute	Weight	New Case	Case in the Case-Base	Local Similarity
Main Symptom	10	Alteration of Sensibility of the superior member (0,4) Pain in the superior Member (0,5)	Elbow pain (0.7)	0.50
Other symptoms	1	Decrease of force in the extending movement	Decrease of force in the extending movement	1.00
Initial Detection	2	Repenting	Progressive	0.00
Concomitant with which activities	4	Tennis (1)	Sports (0.5)	0.25
Improves with which activities	1	To immobilize the superior member (0,6)	Rest during the work (1)	0.30
Worsen with which activities	1	Physical exercises (0.7)	Carrying weight (0.9)	0.40
Familiar antecedents	5	No	No	1.00
Personal antecedents	1	No	No	1.00
Associated pathologies	6	No	No	1.00
Previous treatments	7	No	Physiotherapy (1)	0.00
Work activities	5	Tennis teacher (0.8) Swimming teacher (0.1)	Repetitive movements (1)	0.30
Home activities	1	No	No	1.00
Sports	1	Swimming (0.3) Tennis (1)	Swimming (0.3)	0.43
Emotional suffering	1	No	No	1.00

Palpation Osteo Tendinous	10	Positive	Positive, with irradiation	0.50
Resistive Movement when extending the Fist	5	Positive	Positive	1.00
Elbow passive movements	10	Negative	Positive	0.00
Fist passive movements	9	Negative	Positive	0.00
Passive Prolongation when Extending the Fist	6	Positive	Positive	1.00
Muscular Palpation	8	Positive	Positive	1.00
Resistive Movement	10	Positive	Positive	1.00

Global similarity: (10*0.50 + 1*1.00 + 2*0.00 + 4*0.25 + 1*0.30 + 1*0.40 + 5*1.00 + 1*1.00 + 6*1.00 + 7*0.00 + 5*0.45 + 1*1.00 + 1*0.43 + 1*1.00 + 10*0.50 + 5*1.00 + 10*0.00 + 9*0.00 + 6*1.00 + 8*1.00 + 10*1.00) / (10 + 1 + 2 + 4 + 1 + 1 + 5 + 1 + 6 + 7 + 5 + 1 + 1 + 1 + 10 + 5 + 10 + 9 + 6 + 8 + 10) = 58.38/104 = 0.56

4 Results

For each similarity metric implemented, we initialized the case-base with the same 25 cases, and 15 new cases were used to test the system. This way, the first test case was compared with the 25 initial case-based cases and, after its application, it was inserted in the case-base. The second test case was compared with 26 cases, and so on.

The similarity metrics which retrieved the most similar and useful cases, according with the area specialists, were the Weighted Square Euclidian, which returned the best cases in 8 out of the 15 tests, and the Closest Neighbor (Linear Function), which returned the best case in 7 out of 15 tests. The criteria used to select the best cases was to select the ones with the highest similarity degrees, since their similarity degrees reached at least 0.5.

Figure 2 illustrates the results of the best metrics for this Lateral Epicondilitis system. It illustrates how the retrieved cases match the cases, which were indicated by the area specialists as adequate to be taken into account in the new case. The more cases considered very similar to a particular case they retrieved, the highest their values.

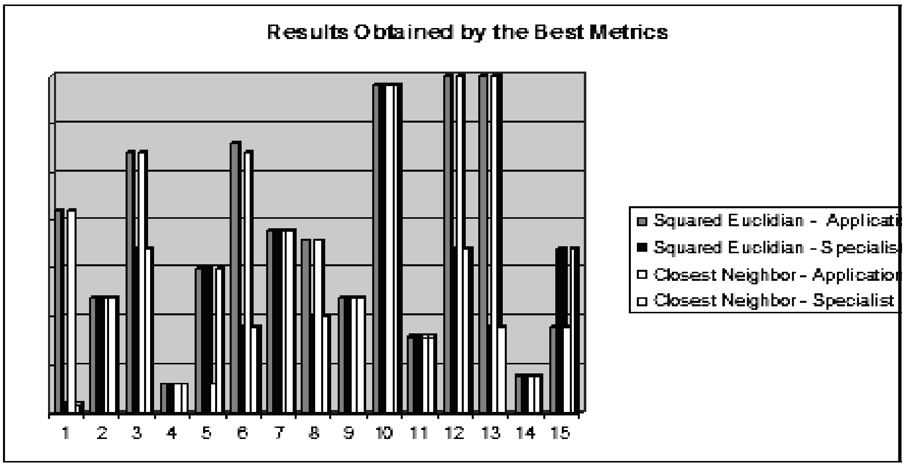


Fig. 2 Similarity metrics results

A reason why these two similarity metrics were particular efficient in the proposed application domain can due to the fact that their calculation included weights for the attributes and values for all the sets of descriptors.

5 Final Remarks

Case-Based Reasoning (CBR) is an Artificial Intelligence technique that simulates the reasoning of a specialist. Its processing is based on reusing past experiences to analyze and propose solutions for a current case.

In such systems, the main knowledge source is the case-base, and the reasoning basically consists in retrieving cases based on their similarity with the current problem, and adapt them to the new situation.

This work presented a Case-Based system applied to Lateral Epicondilitis treatment, which is an elbow tendonitis. The system included the implementation of a set of similarity metrics, which could be tested and compared in the application context. Techniques to deal with multivalued attributes were also proposed and tested. The data concerned real cases collected from public health facilities in Araras and Limeira regions, in the state of Sao Paulo, Brazil. The system will be applied in these regions to help taking better care of the poor people who cannot pay for private health professionals. Currently, the system is been monitored in public health facilities in order to incorporate new cases and to observe and to make the necessary adjustments for its application.

References

1. Aamodt, A., E. Plaza. Case-Based Reasoning: Foundational Issues, methodological Variations, and System Approaches, AICOM, Vol. 7, No. 1 (1994)
2. Camargo, K. Artificial Intelligence Applied to Nutrition. Master Thesis. Santa Catarina Federal University (1999)
3. Defense Advanced Research Projects Agency (DARPA), Case-Based Reasoning, Proceedings of a Workshop on Case-Based Reasoning, Florida (1989)
4. Gresse, C. REMEX - A case based approach for reuse of software measurement experienceware. Proceedings of the 3rd Int. Conference on Case-Based Reasoning, Germany (1999)
5. Julio, M. A Study on Similarity Metrics in Case-Based System Applied to the Health Area. Master Thesis. Univesity of Campinas (2005)
6. Kolodner, J. Case-Based Reasoning. Morgan Kaufmann (1993)
7. Lagemann, G. Client Supporting Using CBR: The Datasul Case. Master Thesis. Santa Catarina Federal University (1997)
8. Lee, R. Intelligent Jurisprudential Research. PhD Thesis. Santa Catarina Federal University (1998)
9. Martins, A. Case-Based Computing: Methodological Contributions to Models of Indexing, Evaliation, Ranking, and Similarity. PhD Thesis. Paraíba Fedreal University (2000)
10. Mello, M. Applying the 7 Project Control Metrics, Rational Software Latin América. White Paper (2002)
11. Osborne, H., Bridge, D. Similarity metrics: A Formal Unification of Cardinal and Non-Cardinal Similarity Measures. Technical Report, Department of Computer Science, University of York (1997)
12. Schank R. Dynamic memory – A theory of reminding and learning in computers and people. Cambridge University Press (1982)
13. Sbrocco J., Freitas, R. Nakamiti, G., Failure Detection on Data Servers Using Intelligent Agents. Proceedings of the VI Brazilian Symposium on Intelligent Automation, Bauru (2003)
14. Turban, E., Jay E. A. Decision Suport Systems and Intelligent Systems, Prentice-Hall, New Jersey (1998)

Providing Intelligent User-Adapted Control Strategies in Building Environments

E. Sierra, R. García-Martínez, A. Hossian, P. Britos and E. Balbuena

Software & Knowledge Engineering Center. Graduate School. Buenos Aires Institute of Technology
Electrotecnic Department. School of Engineering. University of Comahue
Intelligent Systems Laboratory. School of Engineering. University of Buenos Aires

rgm@itba.edu.ar

Abstract. This article describes an intelligent system architecture that based on neural networks, expert systems and negotiating agents technologies is designed to optimize intelligent building's performance. By understanding a building as a dynamic entity capable of adapting itself not only to changing environmental conditions but also to occupant's living habits, high standards of comfort and user satisfaction can be achieved. Results are promising and encourage further research in the field of artificial intelligence applications in building automation systems.

1 Introduction

According to the latest definitions internationally accepted for an "intelligent building", this is a building highly adaptable to the changing conditions of its environment [Krainier, 1996]. But, in an overall concept of comfort, the idea of adaptation to changing environmental conditions may be not enough. Building systems are constructed in order to provide comfortable living conditions for the persons who live in them. It is well known that people usually differ in their personal perceptions of comfort conditions. To some extent, the sensation of comfort is an individual one and it is normally affected by cultural issues. Thus, the idea behind this research is to find techniques based on artificial intelligence in order to provide design recommendations for comfort systems in buildings so that these buildings can also be highly adaptable in terms of the comfort conditions desired by their users. In a few words, a building must "learn" to change its performance not only as a function of environmental conditions, but also as a consequence of preferences set by the people who live in it.

© A. Gelbukh, S. Torres, I. López (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 19, 2006, pp. 235-242

2 The Proposed Intelligent System Architecture

According to the latest trends in the field, intelligence in building systems tends to be distributed [So, 1999].The proposed intelligent system architecture is shown in Figure 1. There is a main computer where the functions of monitoring, visualizing and recording parameters is carried out while the regulation functions are left to the local controllers located throughout the building [Wong, 2001]. These controllers are responsible for taking over local control tasks in the zone they serve. To accomplish its function, the centralized computer contains a database that keeps track of relevant information concerning building user's preferences. For instance, this database keeps records of time, date, number of persons in a room, current temperature and humidity values, as well as temperature and humidity values desired by users. In order to do this, temperature and humidity input panels are located in the different rooms. Each user can eventually set them to what he or she thinks is an ideal comfort condition. As comfort perception is an individual sensation, the database in the main computer keeps track of every individual requirement.

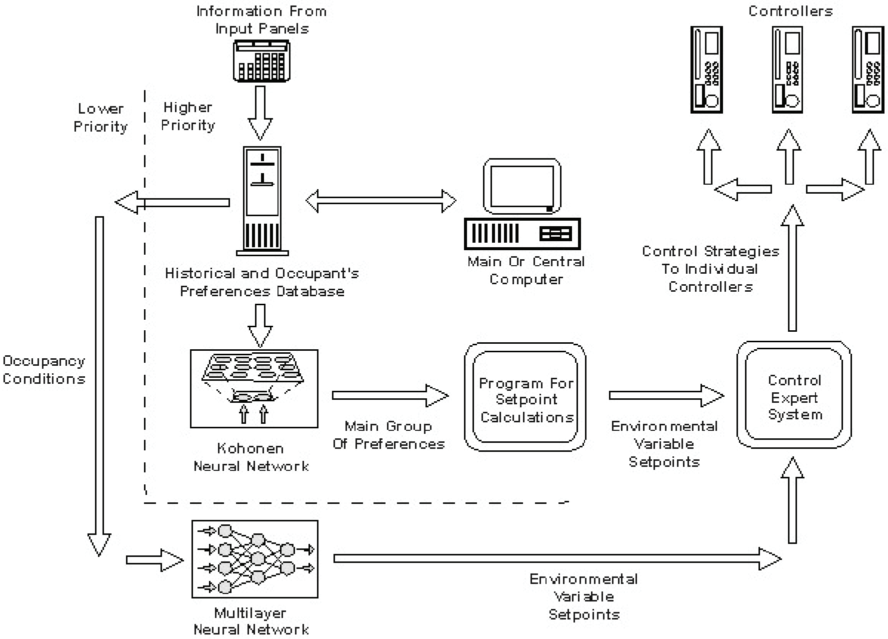


Fig. 1. Intelligent System Architecture

The information contained in the user's requirements database for a given room is applied to a neural network of the self organizational maps of Kohonen (SOM) [Rich & Knight, 1991; Hilera & Martinez, 1995] type, which is used to cluster all the user's requirements and discard all those groups of requirements which are not relevant in terms of their approximation to the main cluster of preferences. Once a unique group of requirements is selected, their values are applied as input to a program which provides the limits as well as the average value for a particular environmental variable. This value is used as reference or set-point for the local control strategies set by an expert system which runs on the main computer. This expert system takes decisions concerning control strategies which are used to activate, deactivate or tune the individual controllers. The information about relevant occupancy and setting conditions, as well as the final values of environmental variables is used to train a multi-layer neural network which outcomes will provide ideal environmental values in case of absence of occupants or of preference information given by them. In any case, set-points assigned to comfort variables provided by the analysis of user's desired environmental conditions is given priority over any automatic calculation of these conditions.

3 Energy Saving Conditions

A very important issue in intelligent buildings technology is related to energy saving policies [Sierra *et al*, 2004]. Optimization procedures carried out to cut off energy consumption rates are not only justified in terms of operation costs reduction but also because of the environmental benefits implied in the adoption of energy saving strategies.

In order to accomplish previously mentioned optimization procedures, an expert system [García-Martinez & Britos, 2004] containing rules that perform energy saving strategies is set up in the central computer. However, it is necessary to verify if the rules defined in the energy saving expert system may eventually alter the comfort conditions established by the control strategy expert system. As it is shown on Figure 2, there is an intelligent negotiation agent [Allen *et al.*, 1991; Conry *et al.*, 1988; Ferber & Drougol, 1992] which runs in the central computer created to determine whether the application of energy saving strategies will: a) not affect current comfort conditions in a given space (not affected) b) affect current comfort conditions but within the limits found by the SOM neural network based upon preference information provided by occupants (partially affected) c) affect current comfort conditions beyond the limits set by occupant's requirements (fully affected).

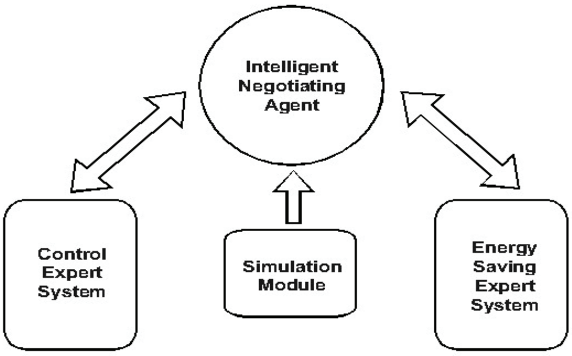


Fig. 2. Negotiation Control and Energy Saving Rules

The policy applied by the intelligent negotiation agent in the different situations mentioned earlier can be summarized as follows:

a) If comfort conditions are not affected, rules defining energy saving strategies are given the highest priority

b) If comfort conditions are partially affected, rules defining energy saving strategies are given an intermediate priority, just lower than the priority given to the rules that regulate the operation of main control actuators.

c) If comfort conditions are fully affected, rules defining energy saving strategies are given the lowest priority.

After the previously described negotiation policy has been applied, the control expert system located in the main central computer has an updated rule base which can be used to set up the operation mode of local controllers (on, off, normal) and tune them accordingly, for example, by determining the appropriate set-point for the control variable.

4 An Example

With the purpose of providing an example that illustrates the functionality of the proposed intelligent system, the operation of the air – handling system depicted in Figure 3 will be described. It is assumed that the HVAC engineer has already designed the air handler in terms of laying out the ductwork, appropriately sizing the fan and heating and cooling coils, and selecting the proper dampers, damper actuators and motor contactor [ASHRAE, 1998; CSI, 1988]. From this design a system diagram has been constructed as shown in Figure 3.

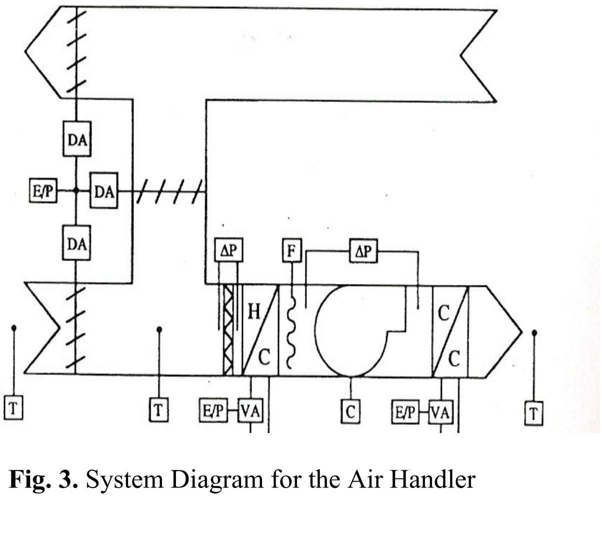


Fig. 3. System Diagram for the Air Handler

The designations DA and VA stand for damper and valve actuators, respectively, C is for electrical contactor and H/C and C/C represent the heating and cooling coils. When building zone building served by the air –handler is "occupied", i.e., the current date and time fall within a certain schedule, the system is said to be in occupied mode. In this mode, the fan is started and the heating and cooling valves and dampers are modulated so as to maintain the set-point temperature in the zone. This is called the "normal" operating condition. Control strategies describe how specific subsystems are to be controlled [IEEE, 1980; NISTIR, 1991]. Thus, some of the rules contained in the rule base of the control expert system will stated as follows:

```
IF    the date and time fall within the specified schedule
THEN  the system shall enter the occupied mode.

IF    the system is in the occupied mode,
THEN  the supply fan shall be turned on,
AND   the normally closed cooling valves and air dampers shall
      be controlled by a sequenced PI (Proportional plus
      Integral) controller to maintain the room air
      temperature set-point to 70 °F.

IF    the date and time fall outside of the specified schedule
AND   the room air temperature exceeds 55 °F
THEN  the system shall enter the unoccupied mode.

IF    the system is in the unoccupied mode
THEN  the supply fan shall be turned off, the heating valve
      shall be set to fully open and the cooling valve and
      outside air dampers shall be set to fully closed.

IF    the date and time fall outside of the specified schedule
AND   the room air temperature is less than or equal to 55 °F,
THEN  the system shall enter setback mode.

IF    the system is in the setback mode,
THEN  the system will remain in this mode until the room air
      temperature exceeds 60 °F.
```

Energy saving strategies were designed in order to diminish energy consumption levels while keeping a satisfying response to the building energy demand profiles. Therefore, some of the rules contained in the rule base of the energy saving expert system can be enunciated in the following manner:

Dry Bulb Economizer Control:

```
IF    the system is in occupied mode
AND   the outside air temperature rises above 65 °F, the dry
      bulb economizer set-point
THEN  the outside air damper will be set to a constant
      position of 20%.
```

Mixed Air Low Limit Control:

```
IF    the system is in occupied mode
AND   the mixed air temperature drops from 40 to 30 °F,
THEN  a proportional (P) control algorithm shall modulate the
      outside air dampers from 100 to 0%. Mixed air low limit
      control shall have priority over dry bulb economizer
      control.
```

Free cooling:

```
IF    the system is in unoccupied mode AND the room air
      temperature exceeds 65 °F
AND   the outside air temperature equals to or is less than 55
      °F,
THEN  the supply fan shall be turned on the heating and
      cooling valves shall be set to fully closed and the
      outside air dampers shall be set to fully open.
```

It is clear that the set-points specified in the previous rules are variables subject to values set by the output of the program which receives as input the outcome of a Kohonen's neural network classifier, as it was sated in section 2, when describing the overall intelligent system architecture. Thus, the system tries to capture the occupants' preferences by modifying the set-points of control variables to users' demands. These new values for set-points are inserted in the rules that perform the control and energy saving strategies of the whole building. These are precisely the rules contained in the expert system that are running in the central computer.

5 Implementation and Results

A prototype of the proposed intelligent system has been implemented in CLIPS, a tool for developing expert systems. Neural network and negotiating agent algorithms have been programmed in C++. The system prototype has been tested in the building of the Ministry of Education, located in the city of Neuquén, Argentina. This building has been designed with a high degree of intelligence. However, its orientation, solar gain controls, surrounding vegetation and other important environmental aspects seem not to have been seriously taken into account in the building's design process. As a result of this, its users show high standards of discomfort, mainly due to the fact that the building is not properly integrated to its surrounding environment. In this scenario, the intelligent software tool proposed in this paper was seen as a starting point for a solution to the building environmental problems. After almost a year of continuous tuning and adjusting procedures, the most updated prototype of the system was put to work. The people who work in this public building was strongly encouraged to set comfort parameters in the input control panels that were installed for this purpose in different building zones. Light, temperature, humidity, safety and energy saving control strategies were supported by the intelligent system located in the building area for environmental control. The comments of users who admitted positive changes in comfort conditions were confirmed by a survey. The survey outcomes were: 75 % percent of users were very satisfied with the performance of the new system, 20 % were just satisfied and 5% not satisfied. Such results encourage advancing in this direction of optimizing the operative and control strategies carried out by the developed system.

6 Conclusions

Techniques of artificial intelligence have been used in many decision, control and automation systems in the last twenty years. Building systems have not been an exception. In this direction, the intelligent system that is proposed in this article tries to contribute in the field of intelligent buildings optimization, by transforming them in a dynamic space, with high standards of comfort and occupant's satisfaction. In this sense, the ability inherent to intelligent systems that are capable of learning from their own environment plays a very important role in the achievement of these building performance optimization goals. Furthermore, results obtained as a consequence of the proposed system implementation are very encouraging. Thus, further research and development work in the field deserves particular attention.

References

1. Krainier, A. 1996. *Toward smart buildings*, Architectural Assn. Graduate School, Environment & Energy Studies Program.
2. So, A. 1999. *Intelligent building systems*, Kluwer Academic Press
3. Wong, K. 2001. *The Intelligent Building Index : ibi manual : version 2.0*, Hong Kong: Asian Institute of Intelligent Buildings
4. Rich Edward and Knight Kevin (1991) *Introduction to Artificial Networks*. Mac Graw-Hill. Publications
5. Hilera J. and Martínez V. 1995. *Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones*. RA-MA, Madrid
6. Sierra, E., Hossian, A., Labriola, C., García Martínez R. 2004. *Optimal Design of Constructions: A Preliminary Model.* Proceedings of the World Renewable Energy Congress (WREC 2004) – Denver, Colorado, Estados Unidos
7. García Martínez, R. and Britos, P. 2004. *Ingeniería de Sistemas Expertos*. Editorial Nueva Librería. 649 páginas. ISBN 987-1104-15-4
8. Allen, J. F, Kautz, H., Pelavin, R. N., and Tenenberg, J.D., 1991. Reasoning About Plans. Morgan Kaufmann Publishers, Inc. San Mateo, California
9. Conry S. E., Meyer R. A., and Lesser V.R., 1988. *Multistage negotiation in distributed planning*. En Bond A and Gasser, L. [Eds] Readings in Distributed Artificial Intelligence. Morgan Kaufmann Publishers, Inc. San Mateo, California
10. Ferber J. and Drougol A., 1992. Using reactive multiagent systems in simulation and problem solving. In Avouris, N.M and Gasser L. [Eds], *Distributed Artificial Intelligence: Theroy and Praxis*. Kluwer Academic Press.
11. ASHRAE. 1989. ASHRAE Guideline 1-1989, *Guideline for Commissioning of HVAC Systems*, Atlanta, ASHARE, 1989.
12. CSI.1988. *CSI Manual of Practi*ce, Alexandria, VA: The Construction Specification Institute.
13. IEEE. 1980. Standard 587-80*, Guide for Surge Voltage in Low Voltage AC Power Circuits*. IEEE. New York.
14. NISTIR. 1991. NISTIR 4606, *Guide Specification for Direct Digital Control Based Building Automation System*, Available from the National Technical Information Service. Springfield. VA 22161.

Light Dimmer Based On Microcontroller PIC16F777

Itzamá López Yáñez and Edgar A. Catalán Salgado

Centro de Investigación en Computación
Juan de Dios Bátiz s/n esq. Miguel Othón de Mendizábal
Unidad Profesional Adolfo López Mateos
Del. Gustavo A. Madero, México, D. F.
México
ilopezb05@sagitario.cic.ipn.mx
ecatalanb05@sagitario.cic.ipn.mx

Abstract. In the present paper we present the design and implementation of a device cappable of mantaining the level of light present in a relatively small area, such as a room or office, in a semi-constant value (ie. with variations which the human eye cannot perceive). This level of light should be defined by the user. In order to achieve this, we propose to use a microcontroller that, according to the difference between the current level of light and the desired level of light, sends a control signal which enables an artificial source of light, such as a lamp, to generate more or less ligth, depending on the cappacities of the light source. The desired level of light should be adjustable in any moment. ...

1 Introduction

The main goal of the current project is to design and implement a device which mantains constant (according to what the naked human eye can perceive) the level of light in a specific area, which is relatively small too, such as a cubicle, room, or office. The former will be achieved by modifying the level of light produced by the artificial light sources present in the area. Also, the desired level of light intended to be mantained must be modifiable by the user at any moment during the working time of the device.

The way chosen to achieve these goals is to employ a microcontroller unit of the PIC16F777 family, built by Microchip^{TM}. This microcontroller will receive data from a light sensor unit, in order to determine the difference, if any, between the current level of light and the desired level of light. With this information, the microcontroller generates a Pulse Width Modulation (PWM) signal which then transmits to a "control" device, which will then allow or disallow the pass of current to the light source according to the PWM signal. As an additional achivement, our proposal should have a low cost, every component should be relatively easy to buy, and the design should be easy to extend.

© A. Gelbukh, S. Torres, I. López (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 19, 2006, pp. 243-251

2 Design

Given the goal of low cost, and in order to simplify and accelerate the design phase, we decided to use a commercial 12 V lamp as the artificial light source and a bipolar transistor of the TIP41C family as the controler device. For the same reason, we choose a 4 MHz oscillator, since it makes the clock cycle of the microcontroller to have a duration of 1 μs, which eases in a great measure the task of calculating times. On the other hand, we want the oscilations in light level to be undetectable by the unaided human eye. Thus, the output signal frequency, which is the frequency at which the light source will work, was restricted to a minimum of 60 Hz, which is the standard for conventional lighting. In regard to the light sensor, there are different available choices, such as digital sensors, photodiodes, phototransistors, and photoresistances. We selected a photoresistance due to its low cost and ease of implementation, both factors in which this option clearly surpasses the other options.

Now, for the microcontroller to be able to know the current level of light, it reads the signal delivered by the light sensor, which is an analog signal, and converts it into a digital value. For this purpose, the Analog–to–Digital Converter (ADC) module was used. Being a 10-bit resolution converter in the PIC16F777 family, this module delivers digital values between 0 and 1023, inclusive. In order to indicate that the lamp must generate more or less light, a control PWM signal was used. This PWM signal codes in the pulse width, how much time of a certain cycle must current be passed to the lamp. Thus, the wider the pulse, the more time is current allowed to pass and the lamp generates light during more time, while at narrower pulses, current passes for less time, making the lamp generate light during less time; and obviously, if the lamp spends more time generating light, then more light is generated. This PWM signal is generated by the PIC's Capture / Compare / PWM (CCP) module, working in its PWM mode. This module, when functioning in PWM mode, has a 10-bit resolution —ie. the Duty Cycle (DC) value of the PWM signal is given between 0 and 1023, inclusive. Thanks to that, the colaboration between the ADC and CCP modules is quite simple, since there is no need for additional manipulation of the values delivered by the ADC module, in order to make them compatible to what the CCP module can manage. The design is shown schematically in Figure 1.

Once the value representing the current level of light has been obtained, it is compared to the value representing the desired level of light. Depending on their difference, the contorl PWM signal is modified, according to the following three possibilities:

- If they are equal, the signal remains unchanged.
- If the current level is less than the desired level, the DC of the PWM signal is increased.
- If the current level is greater than the desired level, the DC of the PWM signal is decreased.

Due to the characteristics of the CCP module in PWM mode, the lowest frequency allowed by a PIC16F777 working with a 4 MHz oscillator is of

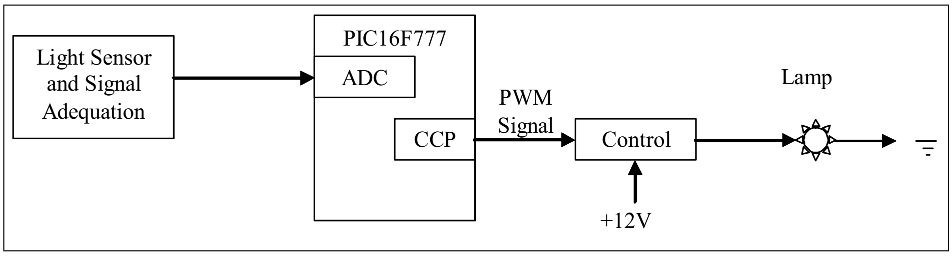


Fig. 1. Block diagram of the proposed design

244.140625 Hz, making the greatest possible PWM cycle of 4.096 ms. Given the restriction on the frecuency of the variations in the light source, established on 60 Hz as inferior limit, 244 Hz perfectly complies with it. The frequency at which the ADC module will sense the signal presented by the light sensor will be of 244.140625 Hz, and will be controled by the CCP module that generates the PWM signal. Since the variations on the general level of light will be given by the variations on the artificial and natural light ources, it is sufficient to sense the level of light at this frequency, since natural light changes at much lower frequencies and the artificial light sources will be controled by the PWM signal, which works at precisely 244 Hz. The level of light which we plan on keeping constant will be defined, originally, during the programming time of the microcontroller, at a value of 25% of the DC. However, during normal operation of the device, it will be possible to modify this value, through a variable resistance or potentiometer, which will change the voltage on a power line between 0 V and 5 V of direc current, which in turn will be converted by an ADC module (different from the one used to convert the signal of the light sensor) to a digital value between 0 and 1023 inclusive.

3 Hardware Implementation

3.1 Input: Light Sensor

As mentioned above, we decided to use a photoresistance as sensor. In this case, we specifically used a 2 MΩ photoresistance since it was the photoresistance with the smallest range we could get at the time. Given that great sensibility and accuracy are not of interest to this design, and the sensing frequency is relatively low, this photoresistance covers the requirements quite well, with the added benefits of being easy to acquire and unexpensive. Now, the signal must be prepared before delivering it to the microcontroller. For this we used a 10 kΩ potentiometer as a calibrator, which delivered a signal between 0.8 V and 5 V. Therefore, the input circuit is shown in Figure 2.

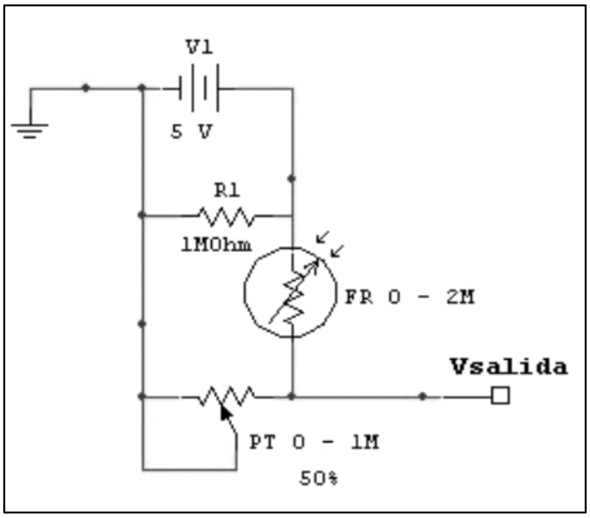


Fig. 2. Diagram of the input circuit

3.2 Output: Light Source

In order to generate the needed level of light, the PWM signal is deliverd to a "control" device, in this case a transistor of the TIP41C family. While the TIP receives a PWM signal with logical value of 1 (5 V), it will let the power signal pass to the lamp; when the value of the PWM signal changes to logical 0 (0 V), the TIP stops feeding power to the lamp. Thus, the pulse width of every cycle of the PWM signal, namely the signal's DC, determines the percentage of the time alloted by the cycle period during wich the lamp is given power. This regulates the average level of voltage delivered to the lamp in a given period of time, making the lamp generate more or less light. The resulting output cicuit is shown in Figure 3. On the other hand, Figure 4 shows the complete diagram, with both input and output circuits, along with the microcontroller and their conections.

4 Software Implementation

In order to obtain the value of the current level of light, compare it with the value of the desired level of light, and then generate the PWM signal according to the difference between both levels, the PIC microcontroller was programmed as described in the flow diagram portrayed in Figure 5.

The ADC module is configured to do the conversion as soon as possible, deliver the converted value and wait for the program to process it. On its part, the PWM module is configured to use the largest period allowed by the PIC when working with a 4 MHz: 4.096 ms. Whenever a PWM cycle ends, the updated value

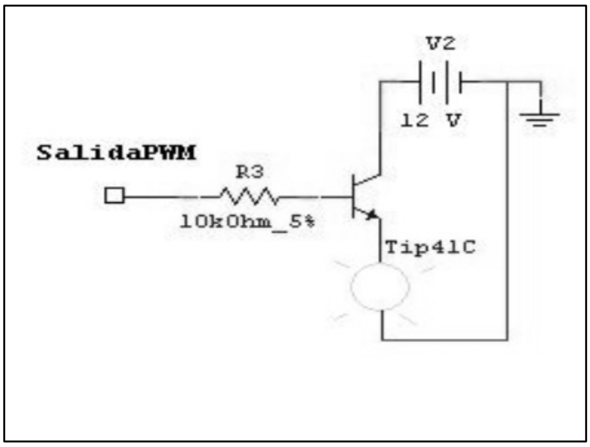


Fig. 3. Diagram of the output circuit

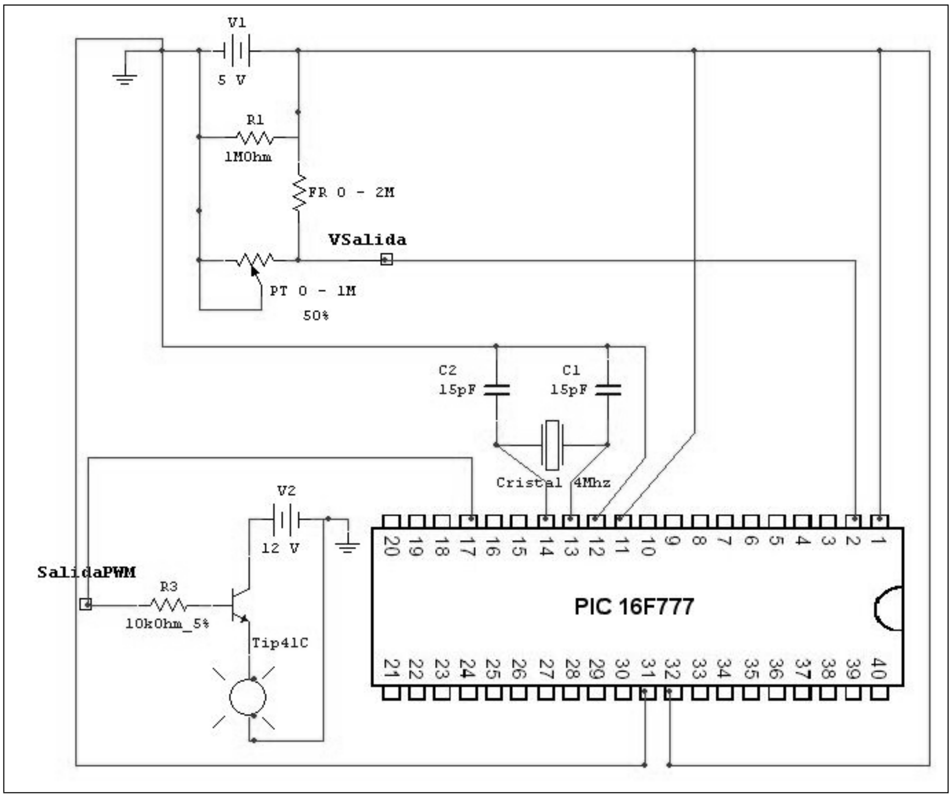


Fig. 4. Complete diagram

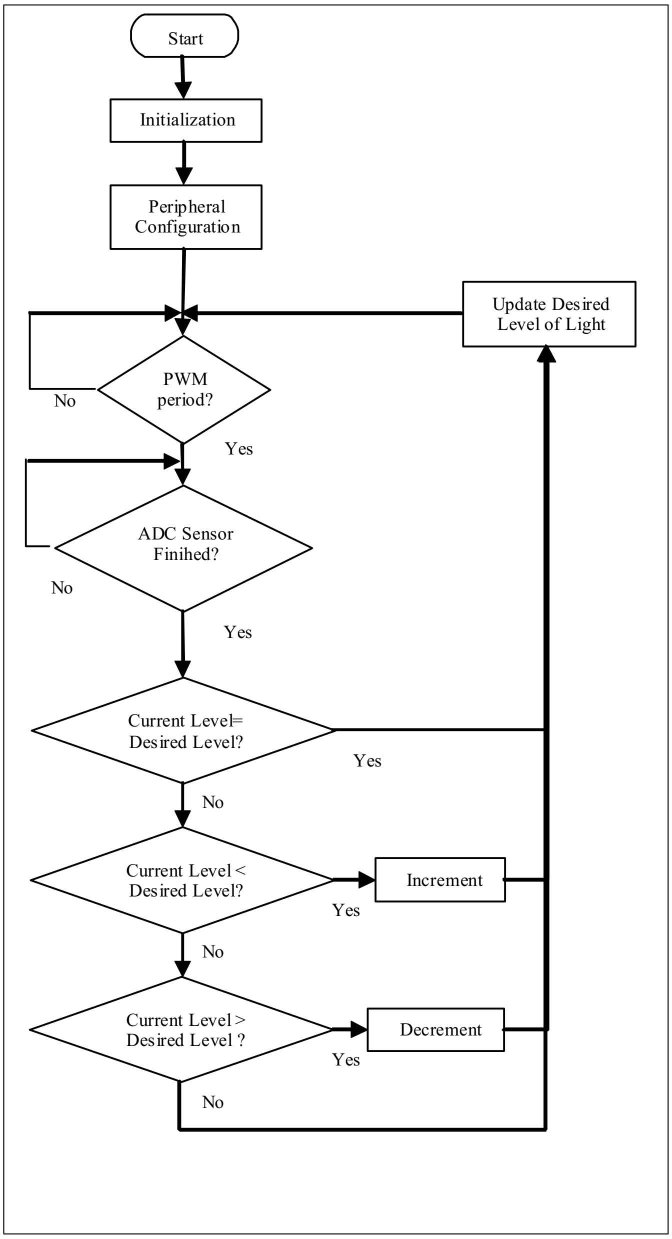

Start
Initialization
Peripheral Configuration
Update Desired Level of Light
PWM period?
No
Yes
ADC Sensor Finihed?
No
Yes
Current Level= Desired Level?
Yes
No
Current Level < Desired Level?
Increment
Yes
No
Current Level > Desired Level ?
Decrement
Yes
No

Fig. 5. Flow diagram of the program

for the DC is latched into the active section, forcing any modification to the DC value to take effect until the next cycle. Also, when the PWM cycle ends, the value delivered by the ADC module is read and written as the current level of light. Before comparing this value to that of the desired level, the ADC is switched to channel 1, in order to make it read the value of the desired level. Now, when comparing the levels of light, both current and desired, there are three mutually exclusive possibilities:

- Both levels are equal.
- The current level is less than the desired level.
- The current level is greater than the desired level.

If they are equal, the value of the desired level remains unchanged. However, if the current level is lower, the DC value is increased by 4. On the other hand, if the current level is higher, the value of the DC is decreased by 4.

After these, the Timer0 module is initialized and the program waits until it completes its cycle to start the next conversion, giving the ADC module enough time to correctly acquire the signal of the next channel. In this channel, the ADC module reads a value between 0 V and 5 V from the signal presented by the pontentiometer, which the user uses to indicate the desired level of light. For this reaon, the Timer0 module is configured to use a prescaler with the value of 4, making its cycle 1.024 ms long. Once the conversion has finilized, the resulting value is stored as the value of the desired level of light and the ADC module is switched back to channel 0. The next conversion, on the ADC module's channel 0, is started when the Timer0 module next finishes its cycle, and then the program waits for the current PWM cycle to end, in order to repeat the whole cycle again.

5 Experimental Results

During the measures taken to the working device, once it was implemented and programmed, two fenomenon were observed, that deviate from the norm. One of them forced us to increase the restrictions on the system, while the other one represents a potential violation to the restriction over variations on the level of light being non observable by the naked human eye.

In the first case, when the light source is receiveng small voltages, meaning something below 20%, the lamp starts to *shudder*. Speaking more cuantitatively and accurately, when the DC falls to percentages of the cycle below 17.5%, a variation of ±5% around the DC value that should be mantained is observed. At higher percentages, such variation is not seen, the value of DC being mantained constant when it should be, varying as it should. This deviation from norm could be unimportant if the frequency at which it happens was high enough to make it imperceptible to the human eye. However, this is not the case, since such frecuency was estimated to be close to 10 Hz; nevertheless, notice that the latter is only an estimate, given that no accurate measure was taken. In order to avoid this deviation, we decided to establish the equivalent to 17.5% of DC as the minimum allowed value for the desired level of light, thus making the available range for this values between 180 and 1023 inclusive.

In the second case, the increments and decrements on the DC have a magnitude of 4. Thus, when there is a severe modification of the desired level of light, such as going from a value of 1023 (sensed signal of 5 V) to a value of 200 in the time of a couple of PWM cycles, say les than 20 ms since one cycle lasts 4 ms, and the desired level of light remains at that second value for a long enough time, as could be 5 s, it takes close to 864 ms for the device to reach the desired DC value. The latter means that, in such an extreme situation, the general level of light would suffer a decrement for more than half a second, time enough for the naked human eye to perceive the change. A possible solution to this problem would be to alter the algorithm with respect to the magnitude of the updates to the DC value, in such way that the time needed for congruence is reduced below the ability of the human eye to detect it, at least below 40 ms, which would be equivalent to a frequency of 30 Hz. Notice, however, that this solution would make the update quite abrupt, which could in turn be undesirable. Another posibility would be to make the update smoother, by making it last longer, perhaps a whole minute, or even more, for a modification as large as that mentioned above.

6 Conclusions

In most situations, the proposed device behaves according to the established goals and restrictions, fullfiling also the objective of being a unexpenisive, easy to implement solution. However, two deviations from these norms were observed. One when the DC falls below 17.5% of the PWM cycle, and the other when the desired level of light is modified by a large amount in a short time. The first problem was solved by making 180 the lowest possible value for the desired level of light, thus disallowing the DC to go below 17.5%. The other problem has not been solved yet. Eventhough these two problems arose, they are not so bad. In the case of the shuddering of the lamp, this phenomenon ressembles a candle. We believe this should be further explored, since it could lead to a unexpensive simulation of candlelight, whithout having to replace the light sources, only installing the device.

Some improvements and future work on the porposed device include:

- Modify the design to allow the use of commercial 120 V lamps.
- Improve the interface for the user to determine the desired level of light. One possibility is to substitute the potentiometer for two buttons, one for increment and one for decrement.
- Add a display where the desired level of light is shown, preferably in some ligh unit, such as lumens, lux, or candelas.
- Add more light sensors. This would allow the device to diferentiate between local and general changes in the level of light, such as shadows or reflexes, and ignore the local ones.

- Add a motion sensor that would allow the device to know when the working area is in use and when it is not. This in turn would allow the device to shut down the light source and even set the microcontroller to sleep mode when the area is not being used, and to restart activity when there is motion detected in the area.
- Design and implement a better HMI, such as a remote control, that includes both the interface for determining the desired level of light and the display of its current value.

References

1. Microchip Technology Inc.: PICmicro^{TM} Mid-Range MCU Family Reference Manual. Microchip Technology Inc. (1997)
2. Microchip Technology Inc.: PIC16F7X7 Data Sheet. 28/40/44-Pin, 8-Bit CMOS Flash Microcontrollers with 10-Bit A/D and nanoWatt Technology. Microchip Technology Inc. (2004)
3. Drix Semiconductor: NPN Silicon Power Transistor TIP41C. Drix Semiconductor
4. Ayala San Martín, G., Luna Ávila, R.: Robotica. Universidad de las Américas Puebla. http://mailweb.udlap.mx/~is114185/Topicos/descripcion.html
5. Universidad Politécnica de Valencia, Escuela Politécnica Superior de Alcoy, Departamento de Ingeniería Eléctrica: Sensores de luz. http://158.42.128.19/asignaturas/LSED/2002-03/Sensores_Luz/fotoconductores.htm
6. Universidad Politécnica de Valencia, Escuela Politécnica Superior de Alcoy, Departamento de Ingeniería Eléctrica: Sensores de luz. http://158.42.128.19/asignaturas/LSED/2002-03/Sensores_Luz/fondo1.htm

Author Index

Índice de autores

Editorial Board of the Volume

Comité editorial de volumen

Additional Reviewers

Árbitros adicionales

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Mayo de 2006.
Printing 500 / Edición 500 ejemplares.