Editorial

THIS issue of Polibits includes ten papers by authors from nine different countries: Brazil, Colombia, Cuba, France, Germany, India, Mexico, Portugal, and Spain. The majority of the papers included in this issue are devoted to such topics as software engineering, automated code generation, cloud computing, analysis of the web, and analysis and retrieval of images.

E. Torres Orue et al. from Cuba in their paper *Process for Unattended Execution of Test Components* describe a methodology to perform software tests. With the increasing complexity of modern software, the quality of the products crucially depends on the correctness and completeness of their testing. While much attention has been paid to totally automatic product verification techniques, in practice they are still far from being applicable, hence the importance of methodologies for manually testing products in a well-organized and systematic way. In this paper, the whole organization of the testing process is addressed, indicating the roles, phases, activities, and artifacts involved in the process. In addition, the authors present a tool that supports the process and allows unattended execution of tests.

G. Vargas-Solar et al. from France, Spain, and Brazil in their paper *Reliable Web Services Composition: An MDD Approach* present an approach for modeling and associating policies to service-based applications. This paper concerns the area of cloud computing, a relatively new software paradigm that receives in the recent years tremendous interest from the industry and user community. Within this paradigm, it is believed that soon we will not use personal computers, including laptops, tablets, etc.—instead, we will send requests to services located at powerful servers and large distributed networks, with the benefits of constant improvement, professional maintenance, reliable and redundant storage, and all this in a manner transparent for the end user. In this paper, the authors propose various extensions for the SOD-M model driven method of service-based computing that improve its reliability and interoperability.

H. Ordoñez et al. from Colombia in their paper *MultiSearchBP: Environment for Search and Clustering of Business Process Models* present an environment for search and clustering of business processes in a multimodal repository of business process models. Their three-level architecture consists of a presentation level, business level, and Storage level and supports operations typical for information retrieval settings. Internally their architecture uses a vector space model to represent the business process models stored in the repository. The authors show that their approach performs well in the experiments.

S. Schnitzer et al. from Germany in their paper *Combining Active and Ensemble Learning for Efficient Classification of Web Documents* address the problem of classifying web documents. With the enormous amount of documents available in Internet nowadays, compact presentation of thematic groups of documents to the user, as well as investigation of the structure of large collections of documents is required for effective access to the information contained in those documents and its adequate use. These processes include classification and grouping of the documents. Usual methods employed for automatic classification of texts involve large text corpora manually marked up by human experts. Development of such corpora is a very slow and expensive process. The authors show how active learning can be employed to minimize human effort by only requiring human intervention when the algorithm cannot reliably perform classification basing on the strategy learnt from its previous interaction with human annotators.

S. Pérez Lovelle et al. from Cuba in their paper *A Proposal to Incorporate More Semantics from Models into Generated Code* consider the problem of reflecting the semantics of software models the code automatically generated from them. Automatic generation of code greatly simplifies and speeds up software development process, as well as reduces the probability of software coding errors. Software development process supported by automatic generation allows the developers to concentrate on important strategic issues: what to develop and not how to code it. The developers' ideas of what is to be developed are specified in formal models, usually expressed by Unified Modeling Language (UML). However, it has been noted that some parts of the semantics of the UML models is lost when automatically converting these models into working code. The authors address the issue of how the semantics of such models is reflected in the code generated by the AndroMDA tool and indicate how it can be improved.

N. Das et al. from India and Portugal in their paper *Comparison of Different Graph Distance Metrics for Semantic Text Based Classification* continue the discussion of text classification, whose importance has just been discussed. They consider semantic approach to the task: documents that are devoted to semantically similar topics are to be clustered together. The clustering process is based on measuring similarity between texts, in this case based on their semantics. There are various commonly used semantic representations, most of them being either based on graphs or equivalent to graphs. However, comparing graphs is computationally expensive. In this paper, the authors show how to reduce the computational complexity of comparing the semantic

representations of texts by using shorter summaries of the texts instead of their full text, which makes the semantic representation simpler and the involved graphs smaller. The authors compare five different graph distance measures.

O. Rodríguez Zalapa et al. from Mexico in their paper *Distance Measurement System using Images to Determine the Position of a Sphere using the XBOX Kinect Sensor* present a method to measure the distance from an image of an object to a given reference point in the same image. This is an important task in computer vision domain. Computer vision has many practical applications, from monitoring of traffic or security to control or autonomous robots and military application. In particular, the existence of truly independent autonomous robots, such as home-helpers or automatic cars, is only possible with high-quality algorithms for visual orientation and, in particular, measuring distances between objects in the images. The authors show that their technique has better accuracy than instruments specifically designed for distance measurements.

V. M. Alonso-Rorís et al. from Spain in their paper *Information Extraction in Semantic, Highly-Structured, and Semi-Structured Web Sources* consider the task of extracting formal and structured information from open web documents, which are very heterogeneous in their nature, with wide variations from totally unstructured texts, images, sounds, and videos to highly structured databases. Better understanding of the nature and relationships of such information sources, as well as of the information they contain, leads to important applications for end users. In this paper, the authors show how the information automatically extracted from open web sources can be employed for two significantly different practical applications: in a recommender system for educational resources on the one hand, and in interactive digital TV applications on the other hand.

L. Flores-Pulido et al. from Mexico in their paper *Computing Polynomial Segmentation through Radial Surface Representation* address the problem of visual information retrieval. While traditional information retrieval operates with texts, in this case both the query and the objects to be retrieved are images. Obviously, this task requires detailed and sometimes sophisticated image-processing techniques, which are the object of the research in this paper. The authors address the problem of constructing appropriate models for representing images in such a way that facilitates their retrieval. The authors show that a modification of the General Principal Component Analysis procedure and other methods based on mathematical operations on the image data lead to high level of performance.

C. M. Zapata Jaramillo et al. from Colombia in their paper *Improving the consistency between textual and graphical syntax of the language of Semat* analize the formal language of the Software Engineering Method and Theory from the area of software engineering. This language has two different representations: as text and as graphical representation similar to block diagrams. The authors observed that the correspondence between these two forms of expression of the same information about software systems can sometimes not be direct, which leads to inconsistencies between these two forms of representation of the language. The authors propose modifications of the textual component of this formal language in order to achieve a more consistent relationship between the textual and graphical forms of the syntax of the core elements of formal language Semat.

This issue of the journal will be useful to researchers, students, and practitioners working in the corresponding areas, as well as to general public interested in advances in computer science and engineering.

Alexander Gelbukh
Editor in Chief

Process for Unattended Execution of Test Components

Emma Torres Orue, Martha D. Delgado Dapena, Jorge Lodos Vigil, and Ezequiel Sevillano Fernandez

***Abstract*—We describe the process to perform software tests. In an enterprise that produces a product line, even if they all have the same goal, they may differ with regard to its development platform, programming language, layer architecture or communication strategies. The process allows standardizing, coordinating and controlling the test execution for all workgroups, no matter their individual characteristics. We present roles, phases, activities and artifacts to address the centralization, reusing and publication of the test scripts and the results of their execution. Additionally, it involves the virtualization for creating test environments, defining steps for its management and publication. Also is presented a tool that supports the process and allow the unattended execution of test components. Finally, we describe two pilot projects demonstrating the applicability of the proposed solution.**

***Index Terms*—Software test process, testing tools, unattended test execution, virtual laboratories.**

I. Introduction

Testing is one of the key activities regarding software quality assurance and quality control. The testing phase should be properly planned and organized in order to prevent errors from manifesting in production and cause undesirable behavior, while minimizing the time and effort employed [1], [2]. Pressman argues that the strategy to test software must provide a map that describes the steps to be taken as part of the test plan, must indicate when they are planned and when these steps will be performed, as well as how much effort, time and resources will be consumed [1]. Several institutions are engaged in the definition of models for software quality [3], [4]. Besides, standards to fulfill the testing process have been designed, for example: IEEE 1008-87 Standard for Software Unit Testing, IEEE 1012-98 Standard for Software Verification and Validation and IEEE 829-98 Standard for Software test Documentation. At the same time, methodologies and processes have been proposed [1], [2], [5], describing activities, roles, and artifacts related to conduct tests within the software development phases.

Manuscript received on February 25, 2013; accepted for publication on May 3, 2013; final version received on June 19, 2014.

Emma Torres Orue, Jorge Lodos Vigil, and Ezequiel Sevillano Fernández are with Segurmatica, Centro Habana, Zanja 651, Havana, Cuba (e-mail: emma@segurmatica.com, lodos@segurmatica.com, ezequiel@segurmatica.com).

Marta D. Delgado Dapena is with the Informatics Studies and Systems Center in the Polytechnic Institute "José Antonio Echevarría," Marianao, 114 Ave. 11909, Havana, Cuba (e-mail: marta@ceis.cujae.edu.cu).

Software product line engineering has received much attention for its potential in the reuse of artifacts throughout the project life cycle [6], [7], [8]. Similarly to the artifacts designed during the implementation, testing artifacts have the same opportunity of being reusable taking into account the similarities identified in the product line [8], [9]. There are some studies related to the generation of test cases and building test scripts from the definition of similarities and variations within a production line [10], [11]. However, scarce references have been found related to the standardization of the execution of the different test components that can be generated in an organization that develops product line.

The automating of the execution of test components is an advantageous aspect in the validation of the elements in a production line [8], [9], [11]. It is suggested by [8] that automation allows artifacts to be tested immediately after being generated and integrated into the system. There are dissimilar solutions to automate generation and execution of test scripts [12], [13], [14], [15]. On the other hand, there are tools to achieve unattended execution of components, from

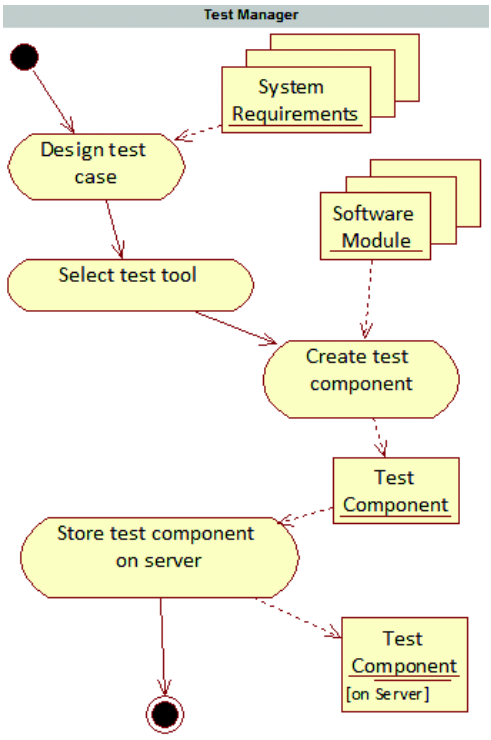


Fig. 1. UML Activity diagram of the phase Test Component Generation.

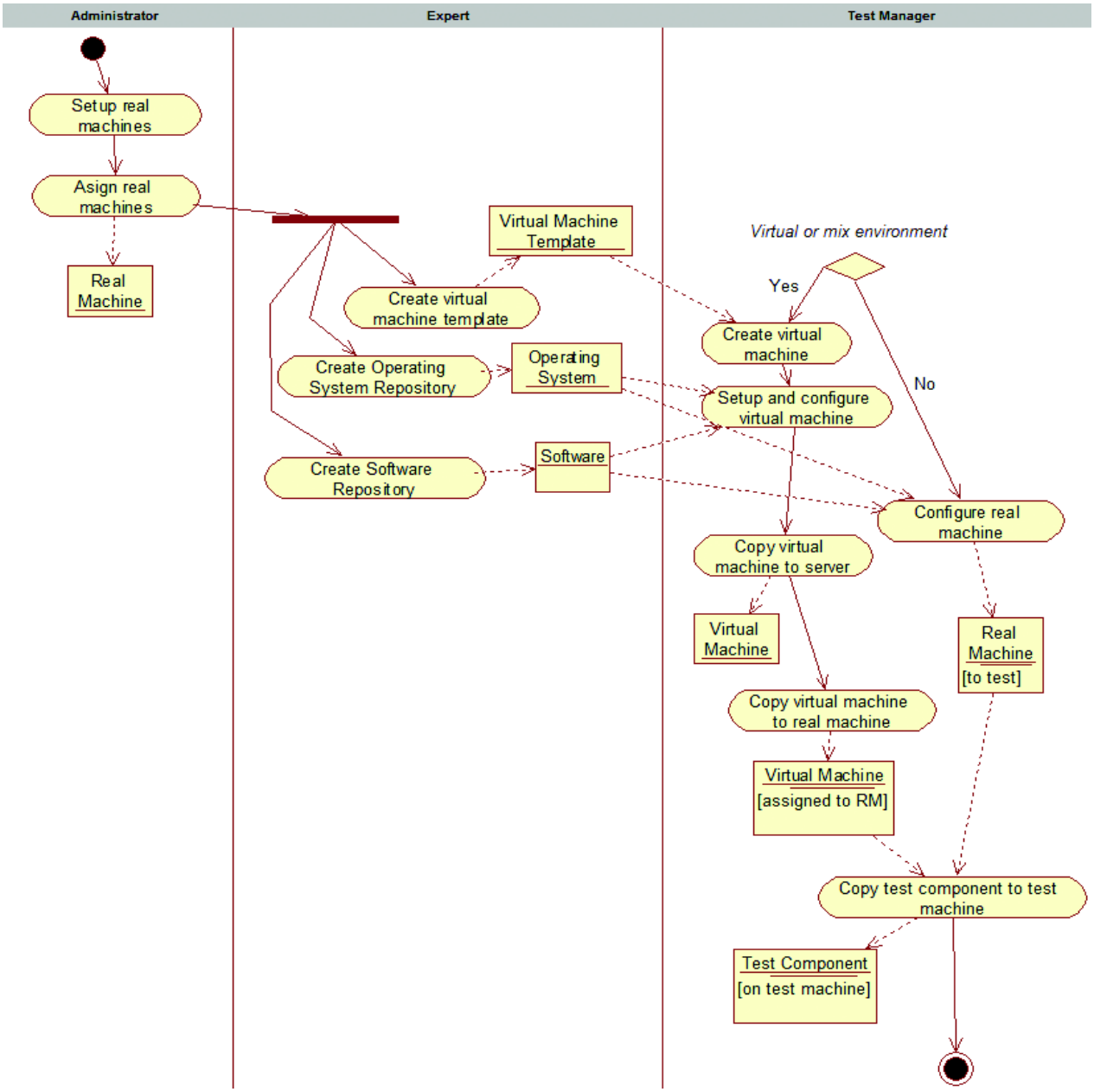


Fig. 2. UML Activity diagram of the phase *Test Environment Creation*.

models, coding scripts languages or test managers with friendly user interfaces [16], [17], [18], [19]. However, these solutions are designed to run scripts generated by specific testing tools. This could implicate the use of different applications for unattended test execution in order to perform all the tests in a product line.

Due to the importance of using large and diverse test environments in order to validate the artifacts in a product line, it has been decided to incorporate the use of virtual machines to facilitate the creation and maintenance of testing laboratories. The integration between virtual laboratory systems [20], [21] and testing tools [17], [18], [22] allows testing applications on virtual machines based on defined test cases. These solutions make possible to record and play back the test runs, as well as to store the results and record conditions that expose the errors for a follow-up. IBM Rational Quality Manager [2] and TestComplete [22] are proprietary solutions offered by the software companies IBM Rational Software and SmartBear respectively. The limitation of Rational's tool is that the test scripts can only be generated by products sold by the company [2], [17]. Furthermore, TestComplete is designed only for machines with Windows operating systems.

This paper describes a process to guide the unattended test execution in real and virtual laboratories. It also shows the characteristics of a tool that enables the execution of scripts

generated by any application, either directly or through a wrapper component. The article presents in the first section the phases and roles involved in the process. The second section shows the automation of the process described using the tool designed to support it. After that, two pilot projects conducted to validate the proposed solution are presented.

II. Process for the Execution of Unattended Tests in Heterogeneous Environments

The process consists of three phases: *Test Component Generation*, *Test Environment Creation* and *Test Execution and Collecting Results*. The inputs are the system modules under test, the system requirements and the associated unit tests. The outputs of the process constitute a knowledge base that stores all the information related to the test runs. Fig. 1 shows a diagram with inputs and outputs of the process. This process is divided in iterations repeated frequently and each one starts from the addition of new modules or modifications to the system in development.

The set of roles in the process includes conventional roles in the stage of software testing such as the Test Manager, the Tester and the Auditor. It also introduces the Expert and Administrator roles. The first one is responsible for making and publishing reusable artifacts for other specialists such as templates, operating systems and software. The second one manages and assigns the computers for developers and test managers, as needed.

A. Phase I: Test Component Generation

The test manager is the one that develops the Component Test artifact. This artifact is a test script generated by a program that automates the validation of one or more functionalities of a system module in different environments. The selected tool must ensure that the script execution results provide as much information as possible. This condition will help discover the source of error in case of failures. Finally, the created component is stored on a server that contains a repository for this artifact. The Fig. 1 displays the activity diagram of this phase.

B. Phase II: Test Environment Creation

The test environment creation involves the Administrator, the Expert and the Test Manager. Fig. 2 shows how the Test Manager prepares test environments based on real machines, assigned by the Administrator, and also based on previously created virtual machine templates and compiled software by the Expert. The testing machines, real or virtual, will have the basic requirements of hardware and software ensuring a successful test run. They must also guarantee the preconditions for the test execution; such as published data, configuration files, among others. Finally, the version of the system under test with its related test components is installed. Both, the templates and virtual machines created, should be stored and published on the server.

C. Phase III: Test Execution and Collecting Results

In the previous phase, the new versions of the software with their associated test components are incorporated. At this stage the associated test components must be run and the results stored. It is recommended to perform this task automatically and unattended, since this would allow the specialist to save time and effort. The results of execution will be stored centrally and will remain public for all specialists involved in the project. Fig. 3 illustrates the activity diagram of this phase.

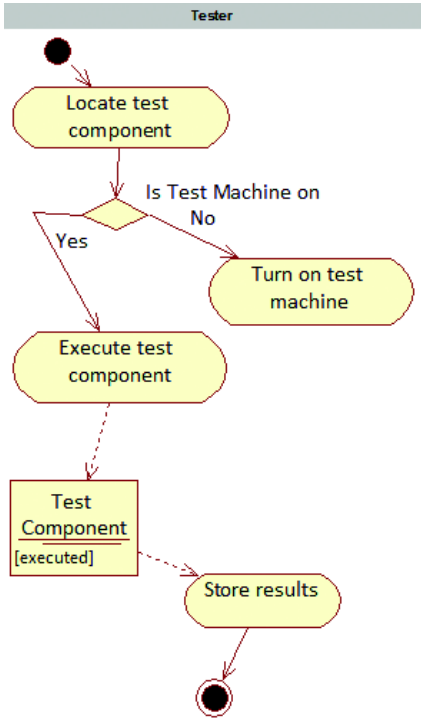


Fig. 3. UML Activity diagram of the phase Test Execution and Collecting Results.

During the implementation of the process, artifacts that record information from the test components, as well as the machine, time and physical path of where they are running, are created. Additionally, the results contain data regarding to the start date, context parameters and user who initiates the process. In Fig. 4 it is shown a class diagram, depicting as entities the repositories obtained during the process.

III. Quality Tool

Quality is a multilayer system developed by the .NET platform. Fig. 5 illustrates the relationship among the system modules. Through a web interface the test components information and its schedules can be recorded. The tool allows defining test suites by grouping test components, which may run simultaneously or sequentially. The Server Web Service provides the functionality to manipulate test labs in real and virtual environments. The Web Interface and the Server Web

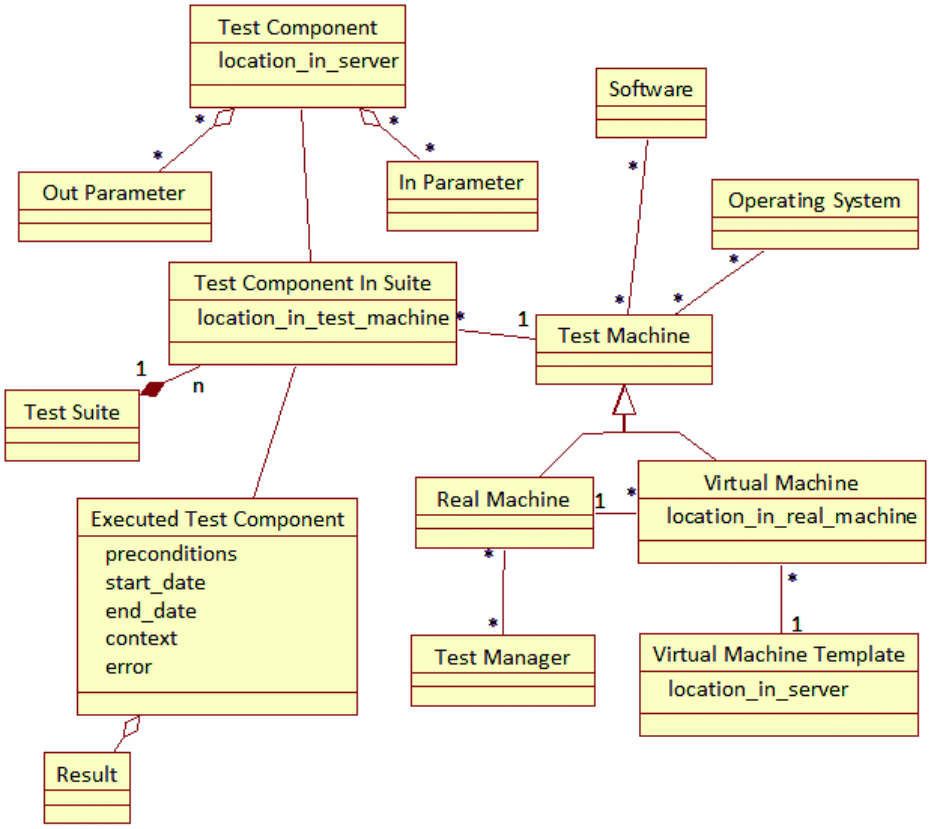


Fig. 4. UML Class Diagram with related entities in the process for running tests unattended.

Service obtain and send data to the database through the library that models the business entities.

During the stage of *Test Execution and Collecting Results*, the Server Web Service determines the real execution machine, and it communicates with the Client Web Service installed on that computer to indicate it to run the test component. According to the configuration, the Client Web Service executes the test script located in a storage device of the real machine or in a virtual machine allocated on it. The scheduled execution is managed by the SQL Agent [23]. The following describes the steps of the process automated with the Quality tool.

The phase *Test Components Generation* is performed with the help of tools available on the market to automate the execution of tests [2], [15], [16], [18], [19]. Communication libraries have been designed to create test scripts that interact with the Quality tool for storing the results and events generated from the runs. It also includes the insertion of the output parameters, which specifies whether the execution was successful or not. In case of error detection, the outputs should reflect the causes.

To handle test components generated by tools that do not allow direct use of the communication library, a wrapper executor has been created. This binary is a test component that's able to run another test script and store the outputs in the Quality System database through the communication library. The input parameters of the wrapper are the test script parameters and the path where the test component is located. Finally, the Test Manager stores the test scripts created in a server repository and inserts into the Quality system interface the scripts names, parameters and locations on the server, as can be seen in Fig. 6.

The first task to start *Test Environment Creation* is performed by the Administrator. He must introduce in the Quality tool the name, operating system and software installed on the physical computers equipped for test execution, as well as the Test Managers who can operate with them. The Expert records information about operating systems, software and virtual machine templates available for create test environments. The Test Manager stores the data of its virtual machines and distributes them to the real machines assigned to it by the Administrator. Fig. 7 captures the data from a real machine and the list of its virtual machines.

During the last stage, test components are organized in test suits to be executed in a particular order. The Test Manager indicates for each script, the test machine (virtual or real) that will execute it. Next, the system makes a copy of the component from the server to the test machine. Components within a test suite may be performed sequentially or simultaneously in one or more test machines. The suite of tests is called Quality Control Process (QCP), and its configuration can be seen in Fig. 8.

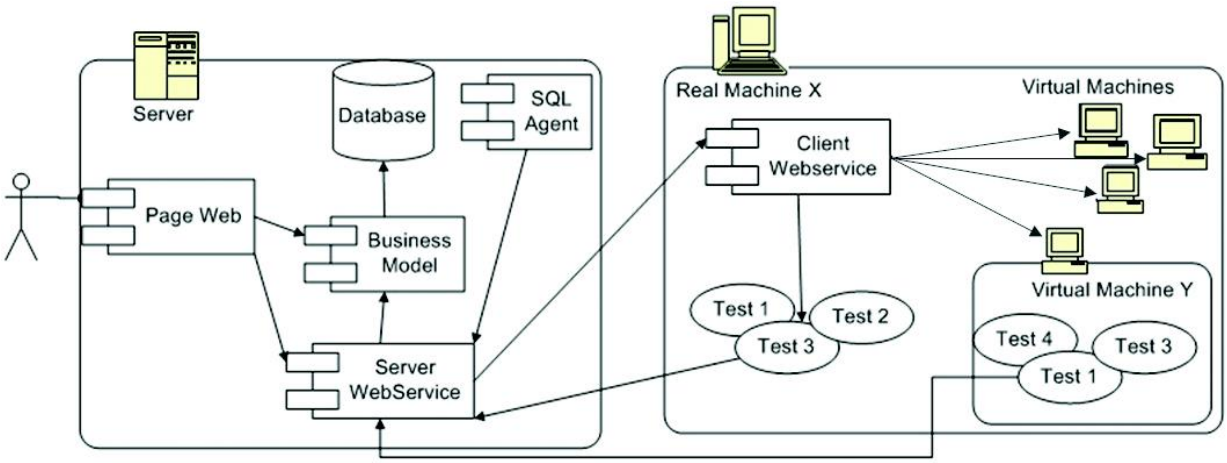


Fig. 5. Internal structure of the Quality Tool.

The input parameters value of the test components can vary for the same test suite, as well as the execution schedules. The definition of these terms is called Instance of the Quality Control Process. This concept allows that one defined test suite can be executed at different times and under different conditions determined by the values of the parameters and the configured test environment. At this point, the required data for a test suite execution is recorded. The Instance of the Quality Control Process can be executed from the web interface by user demand, or on a scheduled date and time.

The Auditor can get the results and events related to every execution of test suite from the Quality web interface. Additionally, during the configuration of the Quality Control Process, Test Managers can identify email addresses to send reports about test execution. These reports can be sent at beginning and/or ending of the execution of a test suite and/or a single test script; likewise if a system error is detected.

IV. Application in Pilot Projects

The designed process was applied to carry out the unattended execution in two pilot projects in a software development company. Those software projects present differences regarding technology, programming language, architecture and team size, demonstrating the applicability of the proposal in various development environments. Results from this experiment demonstrate the improvements that the given solution provides to the test process in the selected areas of the software company.

A. Pilot Project 1

The first pilot project works on a library implemented in C++ native language by a single developer. It is a multiplatform class library designed to extract files from diverse compression formats. Associated with this, there is a test project that contains more than 50 unit tests by format, which were created using the Boost library [24]. These tests validate all the library features. This project was developed and verified prior to the conception of the proposed solution. The developer ran the test project resulting binary in his workstation whenever he wanted to validate it. The library belongs to a system in production at the stations of the company's customers. The process phases performed on the library are explained below.

During the first phase a test script is created using the wrapper provided by the Quality tool. This way, the wrapper execution makes a run on the tests contained in the test project binary and its results are stored in the Quality database. The test input parameter of this component is the relative path where the test project binary is located on the running machine. Later a directory that contains the wrapper and the binary is created on the server. Finally the test component data created is saved in the Quality web interface.

The test environment for the library consists of a real machine. In such a machine Framework .NET 2.0 and IIS 5.1 were installed. Also the tool web service for client machines was published. Subsequently the information related to the test station is stored. The development test environment took a few hours, however this is done only the first time it is introduced into the system.

Through the Quality webpage a test suite composed by the compiled test component is designed. The prepared real machine is selected, indicating the location within the machine where the component will be run. Additionally, the input parameter value representing the relative path of the test project binary is set. The directory containing the test component must be copied from the server to the client machine before execution. The test will be executed with user permission system. Fig. 9 displays the configuration of the test instance for this test component.

The test suite execution takes place every month. For this, one of the schedules configured in the system was selected. The following image shows the description of the selected schedule. After each run, the results are mailed to the library's developer.

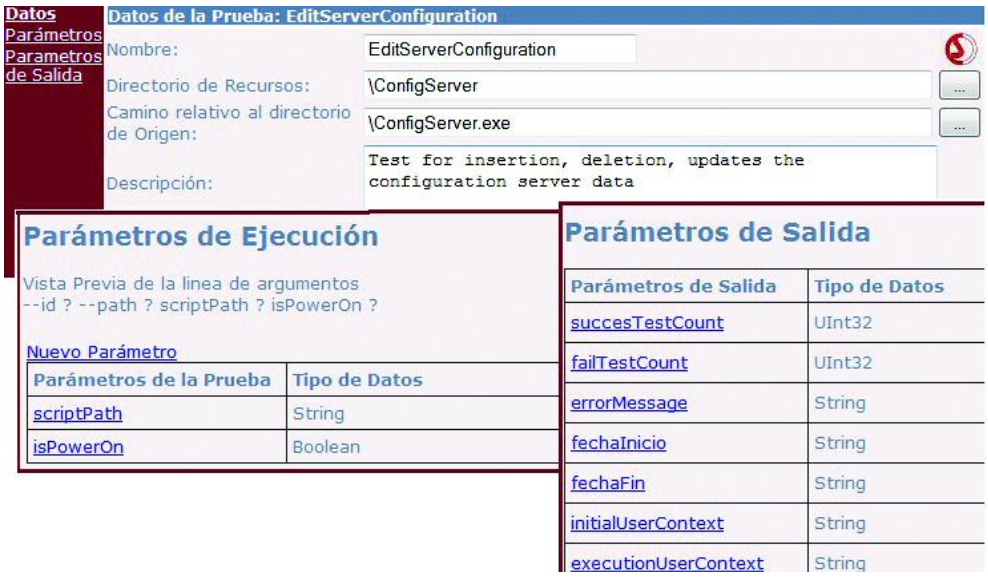


Fig. 6. Test Edition Page of QUALITY tool.

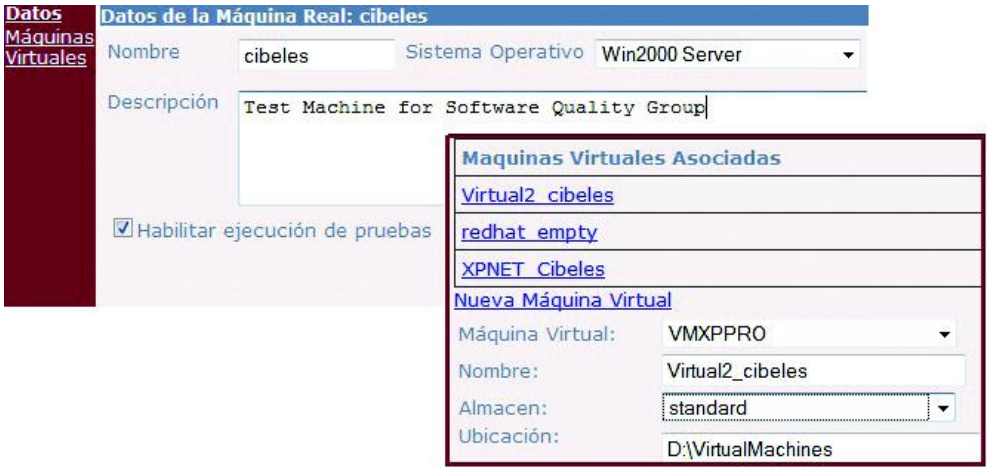


Fig.7. Real and Virtual Machine Association Page of QUALITY tool.

Results obtained in the experiment

The tests run to validate the extraction of the different formats of the library took 49 seconds. In the first test suite execution, 42 errors were detected, especially related to the extraction of files whose formats are less common in the client machines. As the errors detected were solved by the developer, the tool allows to record and report the system progress to other specialists monthly. The recurring execution prevents the introduction of new defects caused by the implementation of additions or modifications, and if this were to happen, the automated process provides a way of finding them in a short time.

B. Pilot Project 2

This pilot project concerned a multilayer system implemented on the .NET platform whose development was in progress at the solution application. It is a distributed system whose architecture consists of a web interface, two web services, and three class libraries. The development team includes internal, temporary internal, and external company developers. As features were added, the developers implemented the related unit tests, which were executed on the workstations. Next, the process applied to this project with the described conditions will be exposed.

In the course of Test Component Generation stage, the Visual Studio Team Suite development tool was used for create scripts. Those test components directly reference the libraries provided by the Quality tool to communicate with the system. Table 1 shows the test types enclosed on the test components created. One component perform one or more validation or verification actions to the system under test, e.g. the Unit Tests script, run all unit tests of a particular system module. For each test, component folders with its files were created on the server. Through the Quality web interface, the information related to these components was stored.

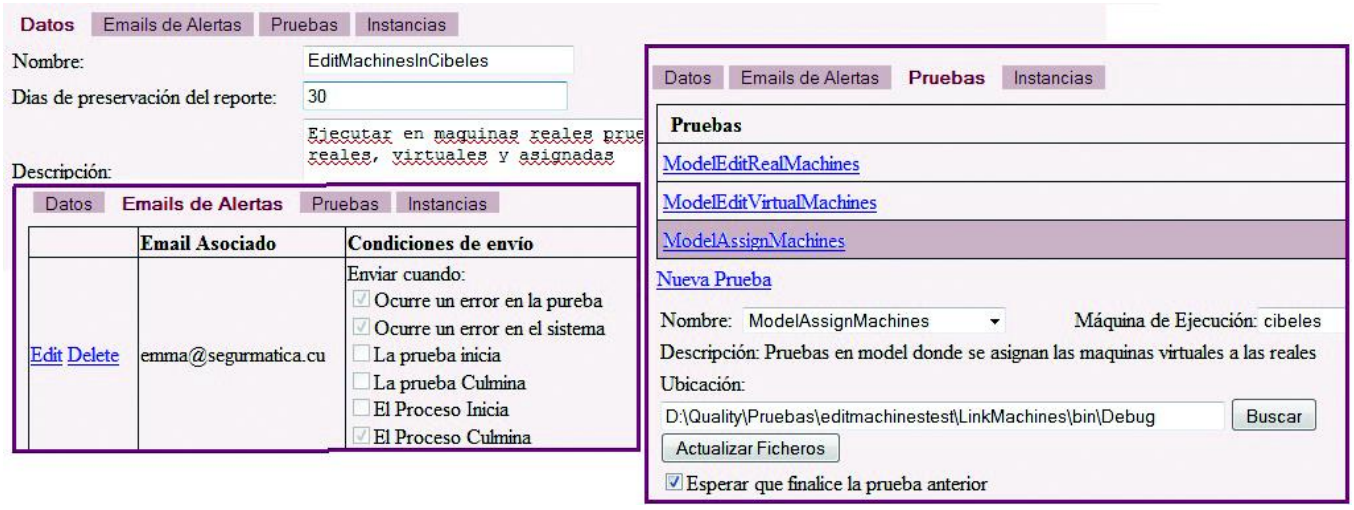


Fig. 8. Quality Control Process Editing Page of QUALITY tool.

Fig. 9. Test Instance Edition Page of QUALITY tool.

The test environment was equipped for two physical machines and four virtual machines. The Fig. 11 shows the test lab established. Real Machine # 1 is intended only to hold three virtual machines, in which test components executions occur. Another virtual machine, fulfill the Server system role, so the tests run on this machine focus on server functions. On this machine the system under test web service for server was installed. Real Machine # 2 plays a dual role, as well as housing the Virtual Machine # 3, run tests on it directly. Because Virtual Machines # 1, 2, 3 and the Real #2 are client machines; the client web service of the system under test was installed on them. The Server and Client machines communicate each other across the local network. Real Machine # 2 coincides with the one built in Pilot Project 1. Therefore, it was not necessary to perform the installation process to incorporate it into the system.

TABLE I.
TEST TYPES DISTRIBUTION BY THE TEST COMPONENTS.

Test Types	Test Component Count
Unit Tests	1
Database Consistency Checking	1
Project Build	1
Functional Tests	20
Integration Tests	5
Web resources availability Checking	1

During the third phase, test scripts are grouped according system functionalities to be verified. Table 2 summarizes the configured tests suite or Quality Control Processes (QCP), the component test types involved and the executions machines. The size of a QCP is expressed as *a/b*, where *a* indicates the

Fig. 10. Selecting a schedule for a test suite execution by Quality tool.

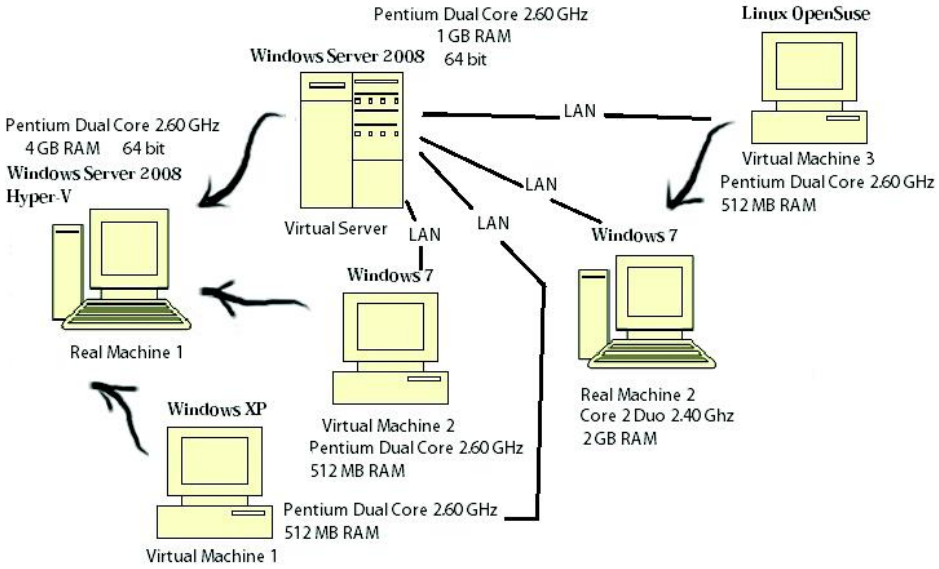


Fig. 11. Test environment for the multilayer system under test.

different test scripts implicated count and *b* represents the number of times the test components are called. For example, the QCP designed to run unit tests consists of three different components: one for Unit Tests, others for Database Consistency Checking and Project Building. Because the system under test consists of 5 core modules, the QCP calls the test scripts 15 times. Another example is the test suite to check the client web service. In this case, there is a test component and it is called four times, for each client machine in the test lab.

Note that the test scripts are reused in the defined QCPs. The same test components have been incorporated into different QCPs to verify diverse aspects of the system under test. This fact is evidenced in the QCPs to verify the on / off virtual machines and to validate test execution in virtual machines. The two test scripts present in the first process are also included in the second, before and after the test component responsible for running the test on the virtual machine.

It was decided to run the test components twice a week, to perform regression testing from changes made during two or three days. Once the configurations for executions were stored on Quality tool, it was possible to update and redistribute the system modules and test components among the test environment machines. After each QCP execution, the results are mailed to the specialists implicated. The following figure shows a fragment of an Instance of the QCP instance run result. The events generated by the test script can be seen.

Results obtained in the experiment

The registered time of execution of all test suites was approximately three hours. It has been estimated that the execution time of all test scripts performed manually takes 7 and a half hours. The possibility to perform runs outside office hours and the presence of an isolated test lab from developer machines saves development and test time for the work team.

The company selected to do the pilot projects, have followed the Scrum Agile methodology. A monthly record or backlog of the development and test work, as well as the defects detected, has been kept. Tasks are planned to be completed in a similar time span, yielding a deliverable

Satisfactorio: EditarUsuariosInWin7

Instancia de Proceso: *EditarUsuariosInMorgan*
Fecha Inicio:8/25/2011 8:52:13 PM
Fecha de culminación:8/25/2011 8:57:29 PM
Id del Reporte: *608*

Instancias de Pruebas ejecutadas

Instancia de Prueba	Estado
EditarInModel_Users	Terminado
EditarInModel_TestManager	Terminado

Descripción	Fecha
Comienzo de la ejecución de la pruebaEditarInModel_Users.	8:52:13
Comienza la ejecución del programa C:\QualityInfo\Tests\EditUsersTest\bin\Debug\EditUsersTest.exe en la maquina Morgan	8:52:29
Se ejecutó exitosamente el script C:\QualityInfo\Tests\EditUsersTest\bin\Debug\ProcesesCtrl_data.sql	8:52:35
Iniciando la prueba	8:52:35
Pruebas de manipulación de contraseñas. Satisfactorias	8:53:01
Pruebas de manipulación de roles. Satisfactorias	8:54:24
Pruebas de inserción y actualización. Satisfactorias	8:55:56
Pruebas de eliminación. Satisfactorias	8:56:17
Terminando la prueba	8:56:17

Fig. 12. Quality report at QCP Instance run finished.

artifact. To complete a functionality it is necessary to perform two or more tasks, depending on its complexity. Tasks are classified according to the stage where they were planned: tasks planned for each sprint during the pregame to develop the system (base tasks) and tasks arising from any errors detected in a previous cycle (defect tasks).

Fig. 13 illustrates the behavior of the tasks performed during one year since March 2010. The graph contains three series, the number of base tasks; the number of defect tasks and the total resulting from the sum of the number of tasks of both previous classifications. The proposed solution was applied in the month of June. In the figure we can see how in the months of June, July and August the number of base tasks increased because specialists created the artifacts required in the defined process. However, in the months of September, October and November there was only a slight increase in the tasks generated by the defects found after the executions of test suites in the previous months.

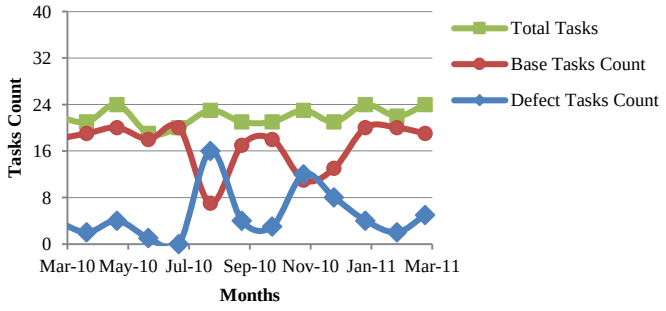


Fig. 13. Graph of the tasks performed before and during insertion of the defined process. Defect tasks are represented by diamonds; base tasks, by circles and the total by squares.

Fig.14 shows the behavior of the work done to develop the pilot project 6 months after that the proposed process was introduced with Quality tool for automation. The graph shows defect tasks have decreased heavily, because the tool helps early detection of errors introduced during implementation. Consequently nonconformities can be discovered and resolved in the same sprint in which they are introduced. From February, the system under test was in the closing stage, for this reason the base tasks also decreased, in turn minimizing the total number of tasks to be performed.

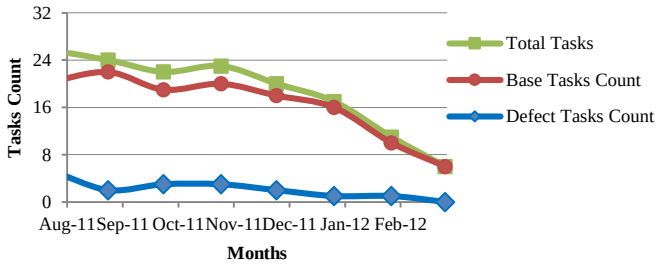


Fig. 14. Graph of the tasks performed 6 months after inserting the proposed process. Defect tasks are represented by diamonds; base tasks, by circles and the total by squares.

Another improvement provided by the proposed solution to the software development process can be seen in the fault detection. The number of errors found was obtained from the backlogs in each sprint. These defects were grouped into two categories: defects detected in functionalities planned in the current sprint (new defects) and failures identified in the current cycle that correspond to functionalities that are considered done in previous cycles (old defects). Fig. 15 describes the conduct of this variable at the same interval of

TABLE II
DEFINED QCPS FOR THE MULTILAYER SYSTEM

Quality Control Process	Size	Test Types Components	Execution Machine
Unit Testing	3/15	Unit Tests	Virtual Machine 1, 2
		Database Consistency Checking	
		Project Build	
Client Web Service Checking	1/4	Web Resources Availability Checking	Virtual Machines 1, 2, 3, Real Machine 2
Virtual Machines Replication	1/1	Integration Tests	Real Machine 2
Machines Edition	3/3	Functional Tests	Virtual Machine 1
Relocate Tests	2/2	Functional Tests	Virtual Machine 1, 2
		Integration Tests	
Timeout expires	1/1	Functional Tests	Virtual Machine 3
Test Edition	3/3	Functional Tests	Machine Real 2
Test Parameters Edition	2/2	Functional Tests	Virtual Machine 2
Process Edition	5/5	Functional Tests	Virtual Machine 1, 2, Real Machine 2
Report Generation	4/4	Functional Tests	Virtual Machine 1
On / off Virtual Machines	2/2	Integration Tests	Real Machine 2
Test run on Virtual Machines	3/3	Integration Tests	Virtual Machine 1
Other operations on virtual machines	1/1	Integration Tests	Real Machine 2
Access Permissions	1/1	Functional Tests	Virtual Machine 1, Real Machine 2
Users Management	2/2	Functional Tests	Virtual Machine 1

Fig. 13 which represents the period before and during implantation of the proposed solution.

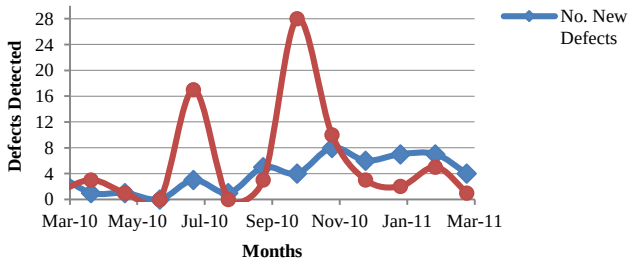


Fig. 15. Graph with the monthly number of defects detected before and during insertion of the proposed process. New defects are represented by diamonds and old defects, by circles.

Above it can be seen that in July, a month after start the solution implantation the system detected an increasing number of defects. However in August few new flaws were found, because the specialists were involved in resolving problems encountered in July, as is outlined in the Fig. 13. In September continues detecting errors due to the creation of new test components. The faults exposed at this stage should have been detected months before the application process. In Fig. 16 can be observed the lines of nonconformities found 6 months after the introduction of the proposed process. At this stage we can see that the number of defects found in features implemented in the current cycle is greater than the number of faults discovered in functionality delivered at earlier sprints.

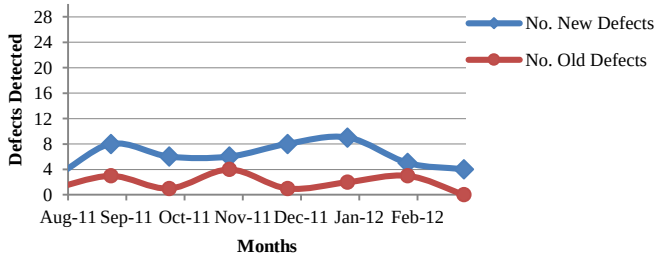


Fig. 16. Graph with monthly the number of defects detected after inserting the proposed process. New defects are represented by diamonds and old defects, by circles.

The most frequent found errors are: Access denied to data; unmanaged exceptions and absent of errors logs, problems with web resources availability, multithreads synchronization issues and timeout expiration. As the errors were detected and solved by developers, unattended executing allows the store and report to stakeholders of the run test results twice a week.

V. CONCLUSION

This paper has detailed a process to standardize the unattended test execution in organizations that develop software product lines. Three stages are defined: Test

Components Generation, Test Environments Creation and Test Execution and Collecting Results. The processes comprise the registration and control of test environments, including machine virtualization. A tool to support the process described has been implemented. This application facilitates the artifacts generation and allows the unattended execution of test components.

The proposed solution adopts the reusability approach proclaimed by the engineering of software product lines. It also promotes the standardization for the test execution of the variations. The process application has reduced the development and testing time, also has provided improvements to the detection of defects in a software company.

ACKNOWLEDGMENTS

This work was supported in part by the Software Security Enterprise, Segurmatica, and the Informatics Studies and Systems Center at the Polytechnic Institute "José Antonio Echevarría." The authors want to thank all workers of the company who participated in the implementation of the proposed solution. We also are very grateful to Heydi Mendez and Annette Morales for their help with translating this paper.

REFERENCES

[1] R. S. Pressman and J. E. Murrieta, *Ingeniería del software, un enfoque práctico*, 6th ed., Mexico, McGraw-Hil Interamericana, 2006, ch.13, pp. 383–414

[2] J. Barnes, *Implementing the IBM® Rational Unified Process® and Solutions: A Guide to Improving Your Software Development Capability and Maturity*. Mexico City, IBM Press, 2007

[3] *Software engineering — Product quality — Part 1: Quality model*, ISO/IEC 9126-1, 2001

[4] R. Pinheiro, K. M. Oliveira, and W. Pereira. "Evaluating the service quality of software providers appraised in CMM / CMMI, *Software Quality Journal*, vol. 17, no. 3, 2009, pp. 283–301; http://link.springer.com/article/10.1007%2Fs11219-008-9065-4

[5] P. Abrahamsson, N. Oza, and M. T. Siponen, "Agile Software Development Methods: A Comparative Review," in *Agile Software Development Current Research and Future Directions*, T. Dingsøyr, T. Dybå and N. Brede, (eds.), Springer, 2010, pp. 31–53

[6] E. Bagheri, F. Ensan, and D. Gasevic, "Decision support for the software product line domain engineering lifecycle," *Automated Software Engineering*, vol. 19, no. 3, 2012 pp. 335–377; link.springer.com/article/10.1007/s10515-011-0099-7

[7] G. K. Hanssen, "Opening Up Software Product Line Engineering," PLEASE'2010 International Workshop, 2010; http://www.idi.ntnu.no/grupper/su/publ/geirkjetil/hanssen-open prodline-please10.pdf

[8] P. A. da Mota Silveira, P. Runeson, I. do C. Machado, E. Santana, S.R. de Lemos, and E. Engstrom, "Testing Software Product Lines," IEEE, vol. 28, no. 5, 2011, pp. 16–20; http://www.computer.org/csdl/mags/so/2011/05/mso2011050016-abs.html

[9] J. Dehlinger and R. R. Lutz, "PLFaultCAT: A Product-Line Software Fault Tree Analysis Tool," *Automated Software Engineering*, vol. 13, no. 1, 2006, pp. 169–193; http://www.cs.iastate.edu/~dehlinge/papers/dehlinger_lutz_AUSE_2006.pdf

[10] A. Bertolino and S. Gnesi, "PLUTO: A Test Methodology for Product Families," *Lecture Notes in Computer Science*, vol. 3014, 2004, pp. 181–197; www.inf.ufpr.br/silvia/topicos/artigostrab10/Bertolino.pdf

[11] E. Uzuncaova, D. Garcia, S. Khurshid, and D. Batory, "Testing Software Product Lines Using Incremental Test Generation," in *Proc. 19th ISSRE*, Washington, DC, 2008, pp. 249–258

[12] A. Edwards, S. Tucker, and B. Demsky, "AFID: an automated approach to collecting software," *Automated Software Engineering*, vol. 17, no. 3, 2010, pp. 347–372. http://link.springer.com/article/10.1007%2Fs10515-010-0068-6#

[13] M. S. Feather and B. Smith, "Automatic Generation of Test Oracles—From Pilot Studies to Application," *Automated Software Engineering*, vol. 8 no. 1, 2001, pp. 31–61. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.7101&rep=rep1&type=pdf

[14] C. Schwarzl and B. Peischl. "Generation of executable test cases based on behavioral UML system models," in *Proc. 5th Workshop on AST '10*, New York, NY, 2010, pp. 31–34

[15] Q. Xie and A. M. Memon, "Designing and comparing automated test oracles for GUI-based software applications," *ACM TOSEM*, vol, 16, no. 1, 2007, pp. 4. http://dl.acm.org/citation.cfm?id=1189752

[16] F. Bouquet, C. Grandpierre, B.Legeard, and F. Peureux, "A Test Generation Solution to Automate Software Testing," in Proc. 3rd International Workshop on AST '08, New York, NY, 2008, pp. 45–48

[17] C. Davis, D. Chirillo, D. Gouveia, F. Saracevic, J. B. Bocarsley, L. Quesada, L. B. Thomas, and M. van Lint, *Software Test Engineering with IBM Rational Functional Tester: The Definitive Resource*, 1st ed. Upper Saddle River, N.J: IBM Press, 2009

[18] J. Levinson, *Software Testing with Visual Studio® 2010*, 1st ed. Redwood City, CA: Addison-Wesley Professional, 2011

[19] L. Chang. "Platform-Independent and Tool-Neutral Test Descriptions for Automated Software Testing," in Proc. ICSE, New York, NY, 2000, pp. 713–715

[20] S. D. Burd, G. Gaillard, E. Rooney, and A. F. Seazzu, "Virtual Computing Laboratories Using VMware Lab Manager," in *Proc. 44th HICSS*, Washington, DC, 2011, pp. 1–9.

[21] J. N. Matthews, T. Deshane, W. Hu, E. M. Dow, J. Bongio, P. F. Wilbur, and B. Johnson. *Running Xen: A Hands-On Guide to the Art of Virtualization*, 1st ed. Upper Saddle River, NJ: Prentice Hall, 2008

[22] N. Rice and S.Trefethen, *TestComplete Version 8 Made Easier: Keyword Testing*, Falafel Software Inc., 2012

[23] R. Walters, G. Fritchey, and C. Taglienti. "Common Database Maintance Tasks," in: *Beginning SQL Server 2008 Administration*, New York, NY: Apress L.P., 2009, pp. 225–233

[24] M. Reddy. "Testing," in *API Design for C++*. Burlington, MA: Morgan Kaufmann (Ed), 2011, ch. 10. pp. 218–328

Reliable Web Services Composition: An MDD Approach

Genoveva Vargas-Solar, Valeria de Castro, Plácido Antonio de Souza Neto, Javier A. Espinosa-Oviedo, Esperanza Marcos, Martin A. Musicante, José-Luis Zechinelli-Martini, and Christine Collet

***Abstract*—This paper presents an approach for modeling and associating *Policies* to services' based applications. It proposes to extend the SOD-M model driven method with (i) the π-SCM, a *Policy* services' composition meta-model for representing non-functional constraints associated to services' based applications; (ii) the π-PEWS meta-model providing guidelines for expressing the composition and the policies; and, (iii) model to model and model to text transformation rules for semi-automatizing the implementation of reliable services' compositions. As will be shown within our environment implementing these meta models and rules, one may represent both systems' cross-cutting aspects (e.g., exception handling for describing what to do when a service is not available, recovery, persistence aspects) and constraints associated to services, that must be respected for using them (e.g., the fact that a service requires an authentication protocol for executing a method).**

***Index Terms*—Methodology, πSOD-M, sevice composition, policy.**

I. INTRODUCTION

SERVICE oriented computing is at the origin of an evolution in the field of software development. An important challenge of service oriented development is to ensure the alignment between IT systems and the business logic. Thus, organizations are seeking for mechanisms to deal with the gap between the systems developed and business needs [1]. The literature stresses the need for methodologies and techniques for service oriented analysis and design, claiming that they are the cornerstone in the development of meaningful services' based applications [2]. In this context, some authors argue that the convergence of model-driven software development, service orientation and better techniques for documenting and improving business processes are the key to make real the idea of rapid, accurate development of software that serves, rather than dictates, software users' goals [3].

Service oriented development methodologies providing models, best practices, and reference architectures to build services' based applications mainly address functional aspects [4], [5], [6], [7], [8]. Non-functional aspects concerning services' and application's "semantics", often expressed as requirements and constraints in general purpose methodologies, are not fully considered or they are added once the application has been implemented in order to ensure some level of reliability (e.g., data privacy, exception handling, atomicity, data persistence). This leads to services' based applications that are partially specified and that are thereby partially compliant with application requirements.

The objective of this work is to model non-functional constraints and associate them to services' based applications early during the services' composition modeling phase. Therefore this paper presents πSOD-M, a model-driven method that extends the SOD-M [6] for building reliable services' based information systems (SIS).

This work (i) proposes to extend the SOD-M [6] method with the notion of *A-Policy* [9] for representing non-functional constraints associated to services' based applications; (ii) defines the π-PEWS meta-model providing guidelines for expressing the composition and the *A-policies*; and finally, (iii) defines model to model transformation rules for generating the π-PEWS model of a reliable services' composition starting from the extended services' composition model; and, model to text transformations for generating the corresponding implementation. As will be shown within our environment implementing these meta models and rules, one may represent both systems' cross-cutting aspects (e.g., exception handling for describing what to do when a service is not available, recovery, persistence aspects) and constraints associated to services, that must be respected for using them (e.g., the fact that a service requires an authentication protocol for executing a method).

Manuscript received on April 24, 2014; accepted for publication on June 12, 2014.

Genoveva Vargas-Solar is with French Council of Scientific Research, LIG-LAFMIA, 681 rue de la Passerelle BP 72, 38402 Saint Martin d'Hères, France (e-mail: Genoveva.Vargas@imag.fr).

Valeria de Castro is with Universidad Rey Juan Carlos, Av Tulipán, Móstoles, Spain (e-mail: Valeria.deCastro@urjc.es).

Plácido Antonio de Souza Neto is with Instituto Federal do Rio Grande do Norte, Av. Senador Salgador Filho, 1559 – Tirol, Natal – RN, Brazil (e-mail: placido.neto@ifrn.edu.br).

Javier A. Espinosa-Oviedo is with French Mexican Laboratory of Informatics and Automatic Control, 681 rue de la Passerelle BP 72, 38402 Saint Martin d'Hères, France (e-mail: javiera.espinosa@gmail.com).

Esperanza Marcos is with Universidad Rey Juan Carlos, Av Tulipán, Móstoles, Spain (e-mail: esperanza.marcos@urjc.es).

Martin A. Musicante is with DIMAp - UFRN, ForAll - Formal Methods and Language Research Laboratory Campus Universitrio – Lagoa Nova, Natal – RN, Brazil (e-mail: mam@dimap.ufrn.br).

José-Luis Zechinelli-Martini is with French-Mexican Laboratory on Informatics and Automatic Control, 681 rue de la Passerelle BP 72, 38402 Saint Martin d'Héres, France (e-mail: joseluis.zechinelli@gmail.com).

Christine Collet is with Grenoble Institute of Technology, Laboratory of Informatics of Grenoble, 681 rue de la Passerelle BP 72, 38402 Saint Martin d'Hères, France (e-mail: Christine.Collet@imag.fr).

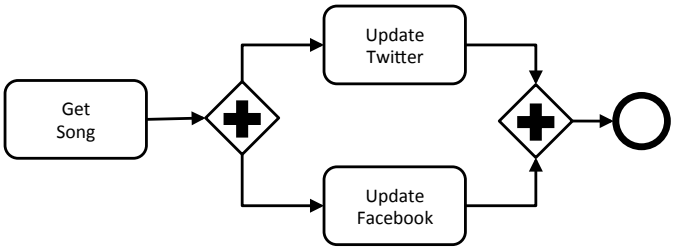


Fig. 1. BPMN model of the "To Publish Music" scenario

The remainder of the paper is organized as follows. Section II gives an overview of our approach. It describes a motivation example that integrates and synchronizes well-known social networks services namely Facebook, Twitter and, Spotify. Sections III, IV, and V describe respectively the three key elements of our proposal, namely the π-SCM and π-PEWS meta-models and the transformation rules that support the semi-automatic generation of reliable services' compositions. Section VI describes implementation and validation issues. Section VII analyses related work concerning policy/contract based programming and, services' composition platforms. Section VIII concludes the paper and discusses future work.

II. Modeling Reliable Services' Compositions with πSOD-M

Consider for instance the following scenario. An organization wants to provide the services' based application "To Publish Music" that monitors the music a person is listening during some periods of time and sends the song title to this person's Twitter and Facebook accounts. Thus, this social network user will have her status synchronized in Twitter and Facebook (i.e., either the same title is published in both accounts or it is not updated) with the title of the music she is listening in Spotify. For developing this services' based application it is necessary to compose the following services calling their exported methods:

- The music service Spotify exports a method for obtaining information about the music a given user is listening:
  - get-Last-Song (userid): String ;
- Facebook and Twitter services export a method for updating the status of a given user:
  - update-Status (userid, new-status): String;

Figure 1 shows the BPMN model[1] of the scenario. The "To Publish Music" scenario starts by contacting the music service Spotify for retrieving the user's musical status (activity Get Song). Twitter and Facebook services are then contacted in parallel for updating the user's status with the corresponding song title (activities Update Twitter and Update Facebook).

Given a set of services with their exported methods known in advance or provided by a service directory, building services' based applications can be a simple task that implies expressing an application logic as a services' composition. The challenge being ensuring the compliance between the specification and the resulting application. Software engineering methods (e.g., [4], [5], [6], [7]) today can help to ensure this compliance, particularly when information systems include several sometimes complex business processes calling Web services or legacy applications exported as services.

A. Modeling a Services' Based Application

Figure 2 shows SOD-M that defines a service oriented approach providing a set of guidelines to build services' based information systems (SIS) [6], [10]. Therefore, SOD-M proposes to use services as first-class objects for the whole process of the SIS development and it follows a Model Driven Architecture (MDA) [11] approach. Extending from the highest level of abstraction of the MDA, SOD-M provides a conceptual structure to: first, capture the system requirements and specification in high-level abstraction models (computation independent models, CIM's); next, starting from such models build platform independent models (PIM's) specifying the system details; next transform such models into platform specific models (PSM's) that bundles the specification of the system with the details of the targeted platform; and finally, serialize such model into the working-code that implements the system.

As shown in Figure 2, the SOD-M model-driven process begins by building the high-level computational independent models and enables specific models for a service platform to be obtained as a result [6]. Referring to the "To Publish Music" application, using SOD-M the designer starts defining an E3value model[2] at the CIM level and then the corresponding models of the PIM are generated leading to a services' composition model (SCM).

Now, consider that besides the services' composition that represents the order in which the services are called for implementing the application "To Publish Music" it is necessary to model other requirements that represent the (i) conditions imposed by services for being contacted, for example the fact the Facebook and Twitter require authentication protocol in order to call their methods for updating the wall; (ii) the conditions stemming from the business rules of the application logic, for example the fact that the walls in Facebook and Twitter must show the same song title and if this is not possible then none of them is updated.

B. Modeling Non-functional Constraints of Services' Based Applications

Adding non-functional requirements and services constraints in the services' composition is a complex task that

[1]Details on BPMN (Business Process Management Notation) can be found in http://www.bpmn.org/

[2]The E3 value model is a business model that represents a business case and allows to understand the environment in which the services' composition will be placed [12].

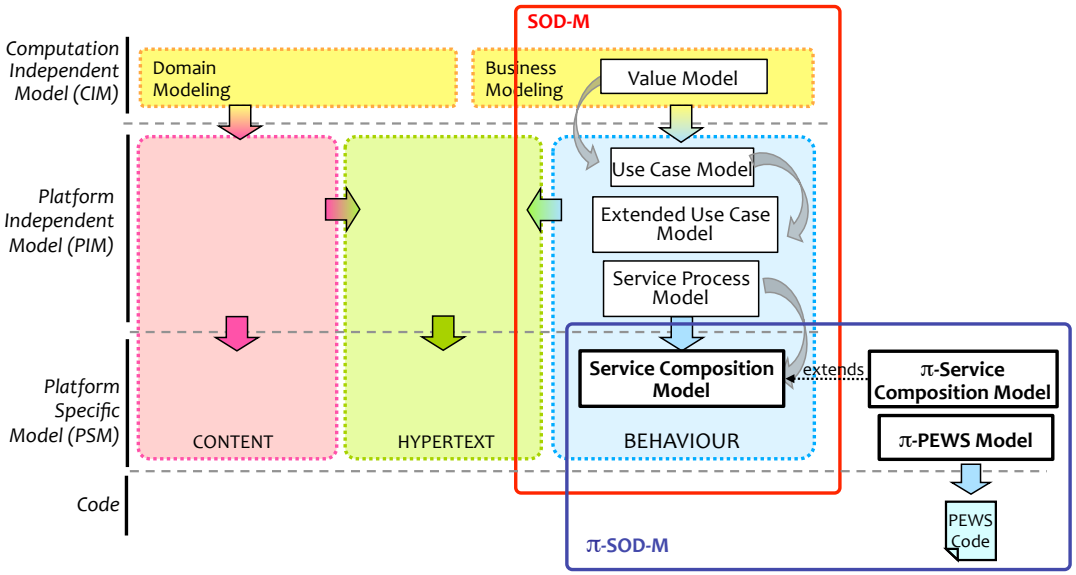


Fig. 2. SOD-M development process

implies programming protocols for instance authentication protocols to call Facebook and Twitter in our example, and atomicity (exception handling and recovery) for ensuring a true synchronization of the song title disseminated in the walls of the user's Facebook and Twitter accounts.

Service oriented computing promotes ease of information systems' construction thanks, for instance, to services' reuse. Yet, this is not applied to non-functional constraints as the ones described previously, because they do not follow in general the same service oriented principle and because they are often not fully considered in the specification process of existing services' oriented development methods. Rather, they are either supposed to be ensured by the underlying execution platform, or they are programmed through ad-hoc protocols. Besides, they are partially or rarely methodologically derived from the application specification, and they are added once the code has been implemented. In consequence, the resulting application does not fully preserve the compliance and reuse expectations provided by the service oriented computing methods.

This work extends SOD-M for building applications by modeling the application logic and its associated non-functional constraints and thereby ensuring the generation of reliable services' composition. As a first step in our approach, and for the sake of simplicity we started modeling non-functional constraints at the PSM level. Thus, in this paper we propose the π-SCM, the services' composition meta-model extended with *A-policies* for modeling non-functional constraints (highlighted in Figure 2 and described in Section III). πSOD-M defines the π-PEWS meta-model providing guidelines for expressing the services' composition and the *A-policies* (see Section IV), and also defines model to model transformation rules for generating π-PEWS models starting from π-SCM models that will support executable code generation (see Section V). Finally, our work defines model to text transformation rules for generating the program that implements both the services' composition and the associated *A-policies* and that is executed by an adapted engine (see Section VI).

III. π-SERVICES' COMPOSITION META-MODEL

The *A-policy* based services' composition meta-model (see in Figure 4) represents a workflow needed to implement a services' composition, identifying those entities that collaborate in the business processes (called BUSINESS COLLABORATORS[3]) and the ACTIONS that they perform. This model is represented by means of a UML activity diagram. Thus, as shown in Figure 3, the meta-model includes typical modeling elements of the activity diagram such as ACTIVITYNODES, INITIALNODES and FINALNODES, DECISIONNODES, etc., along with new elements defined by SOD-M such as BUSINESS COLLABORATORS, SERVICEACTIVITY and ACTION (see the white elements in Figure 4).

- A BUSINESS COLLABORATOR element represents those entities that collaborate in the business processes by performing some of the required actions. They are graphically presented as a partition in the activity diagram. A collaborator can be either internal or external to the system being modelled. When the collaborator of the business is external to the system, the attribute IsExternal[4] of the collaborator is set to true.
- ACTION, a kind of EXECUTABLENODE, are represented in the model as an activity. Each action identified in

[3]We use CAPITALS for referring to meta-models' classes.
[4]We use the sans serif font for referring to models' classes defined using a meta-model.

the model describes a fundamental behaviour unit which represents some type of transformation or processing in the system being modelled. There are two types of actions: i) a WebService (attribute Type is WS); and ii) a simple operation that is not supported by a Web Service, called an ACTIVITYOPERATION (attribute Type is AOP).
– The SERVICEACTIVITY element is a composed activity that must be carried out as part of a business service and is composed of one or more executable nodes.

To illustrate the use of the π-SCM meta-model we used it for defining the *A-policy* based composition model of the "To Publish Music" scenario (see Figure 4). There are three external business collaborators (*Spotify, Twitter* and *Facebook*[5]). It also shows the business process of the "To Publish Music" application that consists of three service activities: *Listen Music*, *Public Music* and *Confirmation*. Note that the action *Publish Music* of the application calls the actions of two service collaborators namely *Facebook* and *Twitter*.

Instead of programming different protocols within the application logic, we propose to include the modeling of non-functional constraints like transactional behaviour, security and adaptability at the early stages of the services' composition engineering process. We model non-functional constraints of services' compositions using the notion of *A-policy* [9], [13], a kind of pattern for specifying *A-policy* types. In order to represent constraints associated to services compositions, we extended the SOD-M services' composition model with two concepts: RULE and A-POLICY (see blue elements in the π-SCM meta-model in Figure 3).

The RULE element represents an event - condition - action rule where the EVENT part represents the moment in which a constraint can be evaluated according to a condition represented by the CONDITION part and the action to be executed for reinforcing it represented by the ACTION part. An *A-policy* groups a set of rules. It describes global variables and operations that can be shared by the rules and that can be used for expressing their Event and Condition parts. An *A-Policy* is associated to the elements BUSINESSCOLLABORATOR, SERVICEACTIVITY and, ACTION of the π-SCM meta-model (see Figure 3).

Given that *Facebook* and *Twitter* services require authentication protocols in order to execute methods that will read and update the users' space. A call to such services must be part of the authentication protocol required by these services. In the example we associate two authentication policies, one for the open authentication protocol, represented by the class Twitter OAuthPolicy that will be associated to the activity UpdateTwitter (see Figure 4). In the same way, the class Facebook HTTPAuthPolicy, for the http authentication protocol will be associated to the activity UpdateFacebook. OAuth implements the open authentication protocol. As shown in Figure 4, the *A-policy* has a variable Token that will be used to store the authentication token provided by the service. This variable type is imported through the library OAuth.Token. The *A-policy* defines two rules, both can be triggered by events of type ActivityPrepared: (i) if no token has been associated to the variable token, stated in by the condition of rule R_1, then a token is obtained (action part of R_1); (ii) if the token has expired, stated in the condition of rule R_2, then it is renewed (action part of R_2). Note that the code in the actions profits from the imported OAuth.Token for transparently obtaining or renewing a token from a third party.

HTTP-Auth implements the HTTP-Auth protocol. As shown in Figure 4, the *A-policy* imports an http protocol library and it has two variables username and password. The event of type ActivityPrepared is the triggering event of the rule R_1. On the notification of an event of that type, a credential is obtained using the username and password values. The object storing the credential is associated to the scope, i.e., the activity that will then use it for executing the method call.

Thanks to rules and policies it is possible to model and associate non-functional properties to services' compositions and then generate the code. For example, the atomic integration of information retrieved from different social network services, automatic generation of an integrated view of the operations executed in different social networks or for providing security in the communication channel when the payment service is called.

Back to the definition process of a SIS, once the *A-policy* based services' composition model has been defined, then it can be transformed into a model (i.e., π-PEWS model) that can support then executable code generation. The following section describes the π-PEWS meta-model that supports this representation.

IV. π-PEWS META-MODEL

The idea of the π-PEWS meta-model is based on the services' composition approach provided by the language PEWS [14], [15] (*Path Expressions for Web Services*), a programming language that lets the service designer combine the methods or subprograms that implement each operation of a service, in order to achieve the desired application logic. Figure 5 presents the π-PEWS meta-model consisting of classes representing:

– A services' composition: NAMESPACE representing the interface exported by a service, OPERATION that represents a call to a service method, COMPOSITEOPERATION, and OPERATOR for representing a services' composition and PATH representing a services' composition. A PATH can be an OPERATION or a COMPOUND OPERATION denoted by an identifier. A COMPOUND OPERATION is defined using an OPERATOR that can be represent sequential (.) and parallel ($\|$) composition of services, choice ($+$) among services, the sequential ($*$) and parallel ($\{\ldots\}$) repetition of an

[5]We use *italics* to refer to concrete values of the classes of a model that are derived from the classes of a meta-model.

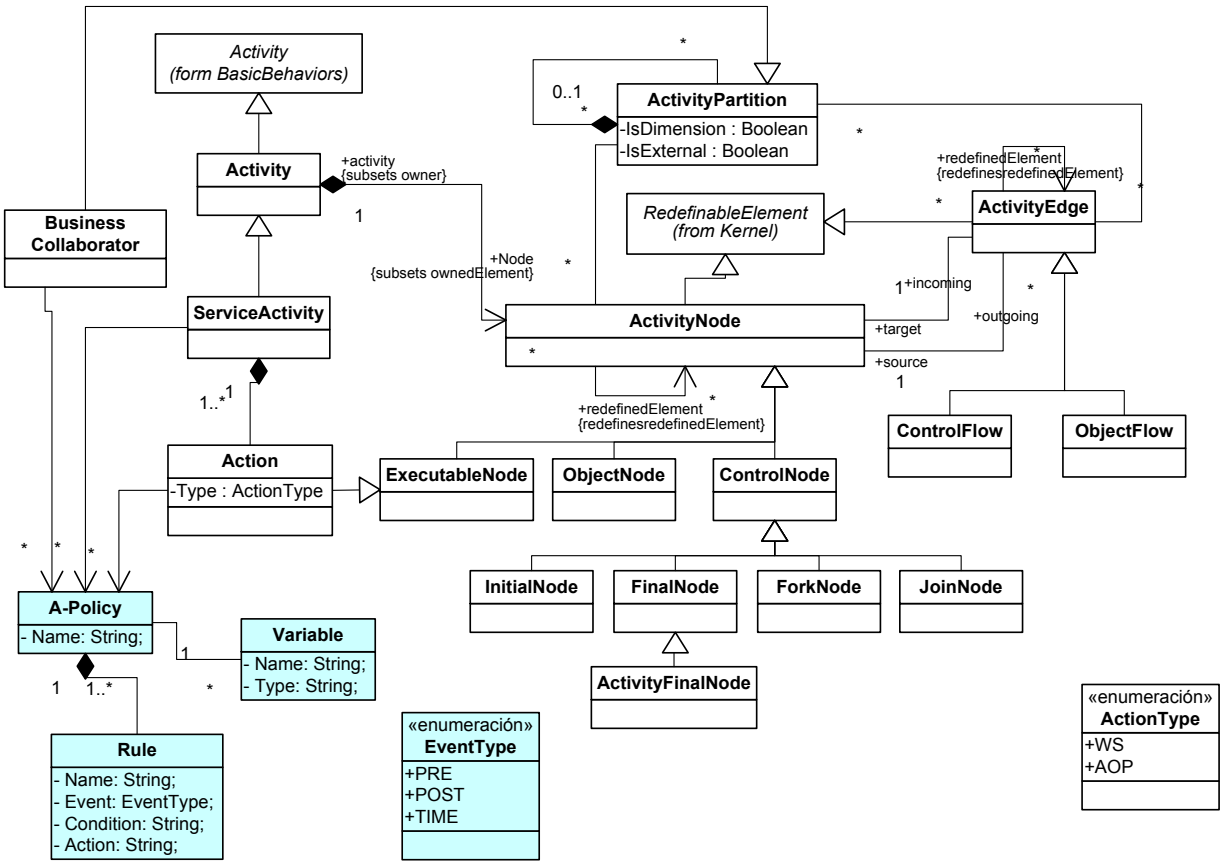


Fig. 3. *A-policy* based services' composition meta-model (π-SCM)

operation or the conditional execution of an operation ($[C]S$).

– *A-Policies* that can be associated to a services' composition: A-POLICY, RULE, EVENT, CONDITION, ACTION, STATE, and SCOPE.

As shown in the diagram an A-POLICY is applied to a SCOPE that can be either an OPERATION (e.g., an authentication protocol associated to a method exported by a service), an OPERATOR (e.g., a temporal constraint associated to a sequence of operators, the authorized delay between reading a song title in Spotify and updating the walls must be less then 30 seconds), and a PATH (e.g., executing the walls' update under a strict atomicity protocol – all or noting). It groups a set of ECA rules, each rule having a classic semantics, i.e, *when an event of type E occurs if condition C is verified then execute the action A*. Thus, an *A-policy* represents a set of reactions to be possibly executed if one or several triggering events of its rules are notified.

– The class SCOPE represents any element of a services' composition (i.e., operation, operator, path).
– The class A-POLICY represents a recovery strategy implemented by ECA rules of the form EVENT - CONDITION - ACTION. A *A-policy* has variables that represent the view of the execution state of its associated scope, that is required for executing the rules. The value of a variable is represented using the type VARIABLE. The class A-POLICY is specialized for defining specific constraints, for instance authentication *A-policies*.

Given a π-SCM model of a specific services' based application (expressed according to the π-SCM meta-model), it is possible to generate its corresponding π-PEWS model thanks to transformation rules. The following section describes the transformation rules between the π-SCM and π-PEWS meta-models of our method.

V. TRANSFORMATION RULES

Table I shows the transformation principle between the elements of the π-SCM meta-model used for representing the services' composition into the elements of the π-PEWS meta-model. There are two groups of rules: those that transform services' composition elements of the π-SCM to π-PEWS meta-models elements; and those that transform rules grouped by policies into *A-policy* types.

A. Transformation of the Services' Composition Elements of the π-SCM to the π-PEWS Elements

A named action of the π-SCM represented by *Action* and *Action:name* is transformed to a named class OPERATION

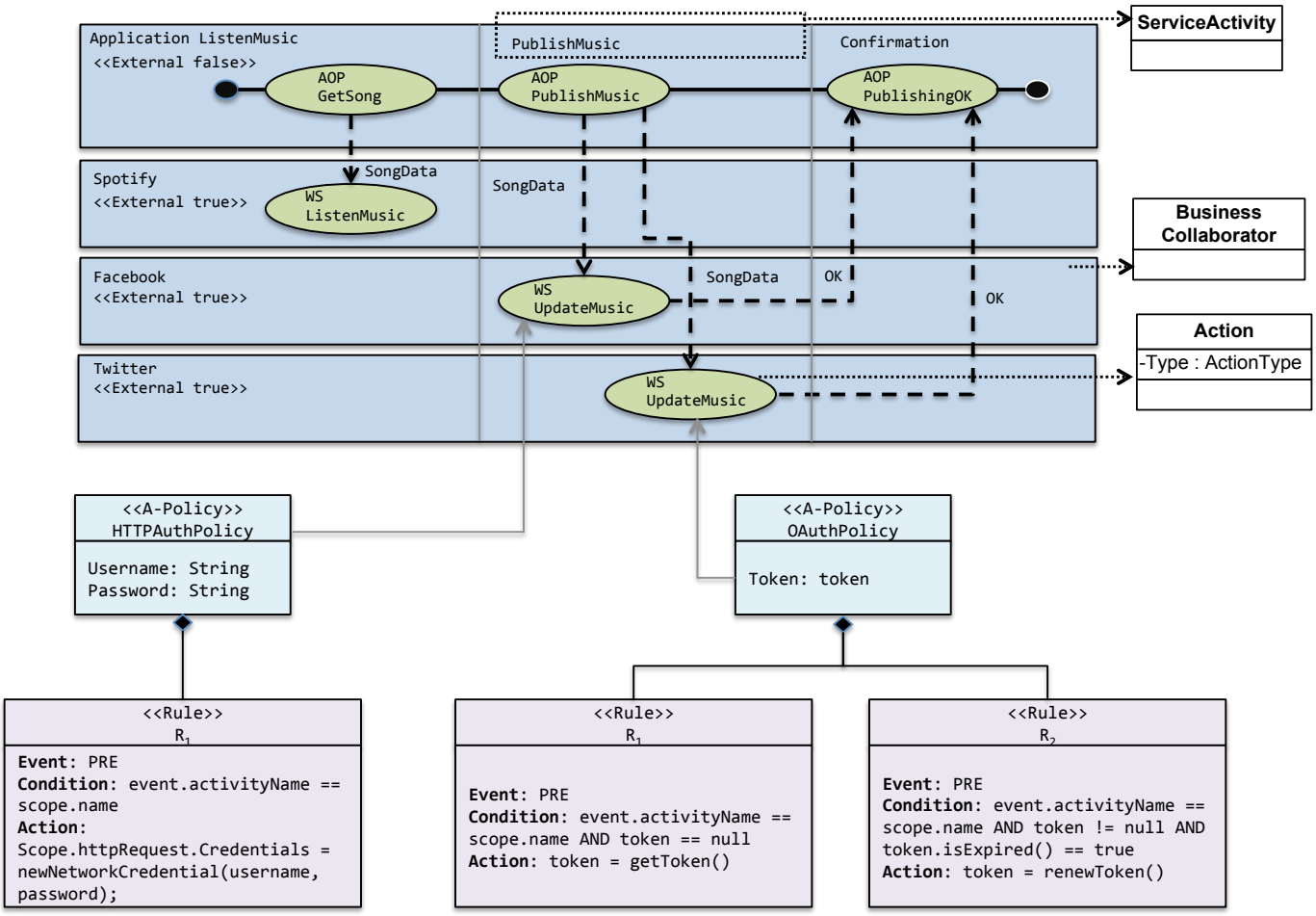


Fig. 4. Services' composition model for the business service "To publish music"

with a corresponding attribute name OPERATION:NAME. A named service activity represented by the elements *ServiceActivity* and *ServiceActivity:name* of the π-SCM, are transformed into a named operation of the π-PEWS represented by the elements COMPOSITEOPERATION and COMPOSITEOPERATION:NAME. When more than one action is called, according to the following composition patterns expressed using the operators *merge, decision, fork and join* in the π-SCM the corresponding transformations, according to the PEWS operators presented above, are (see details in Table I):

- $op_1.op_2$ if no *ControlNode* is specified
- $(op_1 \parallel op_2).op_3$ if control nodes of type *fork, join* are combined
- $(op_1 + op_2).op_3$ if control nodes of type *decision, merge* are combined

In the scenario "To Publish Music" the service activity PublishMusic of the π-SC model specifies calls to two Activities of type *UpdateMusic*, respectively concerning the Business Services *Facebook* and *Twitter*. Given that no ConstrolNode is specified by the π-SC model, the corresponding transformation is the expression that defines a Composite Operation named *PublishSong* of the π-PEWS model of the form (PublishFacebook $\parallel$ PublishTwitter).

B. Transformation of Rules Grouped by A-policies in the π-SCM to A-Policies of π-PEWS

The *A-policies* defined for the elements of the π-SCM are transformed into A-POLICY classes, named according to the names expressed in the source model. The transformation of the rules expressed in the π-SCM is guided by the event types associated to these rules. The variables associated to an *A-policy* expressed in the π-SCM as $<$*Variable:name, Variable:type*$>$ are transformed into elements of type VARIABLE with attributes NAME and TYPE directly specified from the elements *Variable:name* and *Variable:type* of the π-SCM model.

As shown in Table I, for an event of type *Pre* the corresponding transformed rule is of type PRECONDITION; for an event of type *Post* the corresponding transformed rule is of type POSTCONDITION; finally, for an event of type *TimeRestriction* the corresponding transformed rule

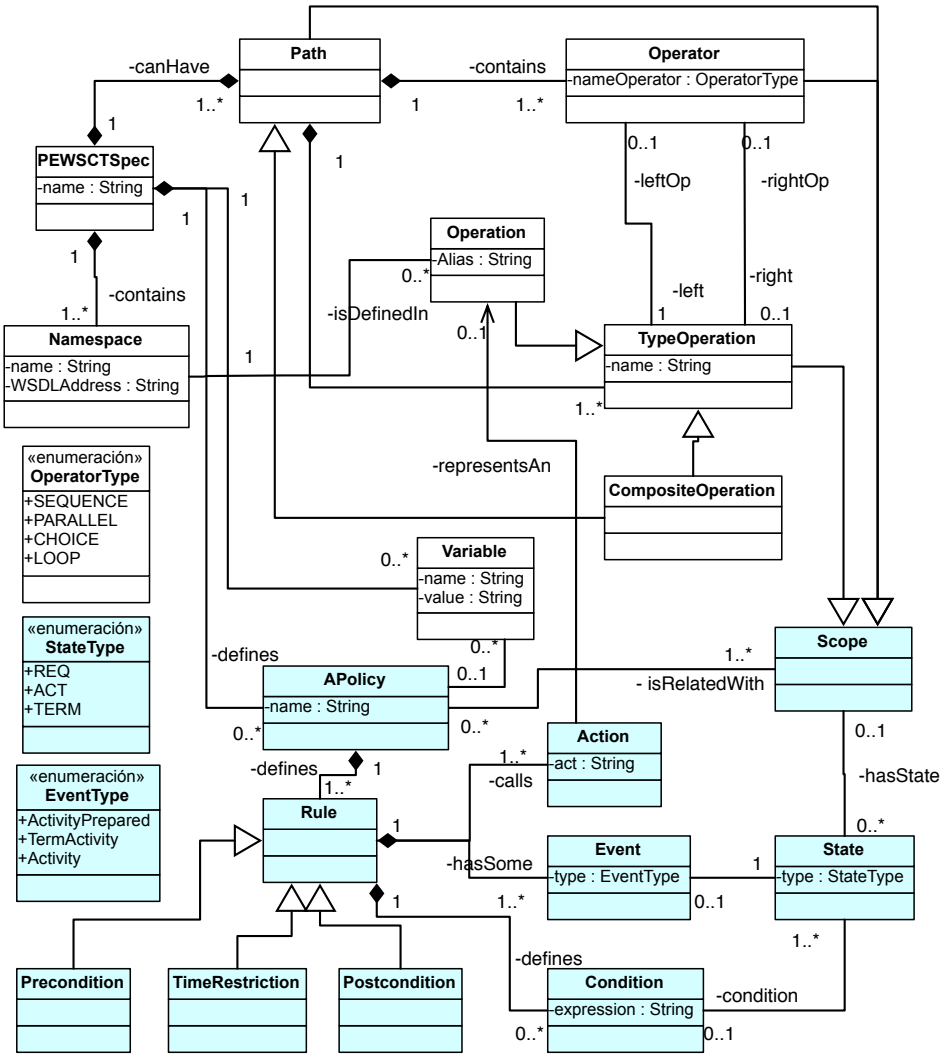


Fig. 5. π-PEWS Metamodel

is of type TIME. The condition expression of a rule in the π-SCM (*Rule:condition*) is transformed into a class *Condition:expression* where the attributes of the expression are transformed into elements of type ATTRIBUTE.

In the scenario "To Publish Music" the Policies *OAuthPolicy* and *HTTPAuthPolicy* of the π-SCM model are transformed into *A-policies* of type Precondition of the π-PEWS model of the scenario. Thus in both cases the events are of type ActivityPrepared. These policies, as stated in the π-SCM model, are associated to Activities. In the corresponding transformation they are associated to Operations *PublishFacebook* and *PublishTwitter*.

VI. IMPLEMENTATION ISSUES

This section describes the πSOD-M development environment that implements the generation of *A-policies*' based services' compositions. For a given services' based application, the process consists in generating the code starting from a π-SCM modeling an application. Note that the services' composition model is not modeled from scratch, but it is the result of a general process defined by the πSOD-M method in which a set of models are built following a service oriented approach [6].

A. πSOD-M Development Environment

Figure 6 depicts a general architecture of the πSOD-M Development Environment showing the set of plug-ins developed in order to implement it. The environment implements the abstract architecture shown in Figure 2. Thus, it consists of plug-ins implementing the π-SCM and π-PEWS meta-models used for defining models specifying services' compositions and their associated policies; and ATL rules for transforming PSM models (model to model transformation) and finally generating code (model to text transformation).

TABLE I
TRANSFORMATION RULES: FROM π-SERVICECOMPOSITION TO π-PEWS

Source: π-SCM	Mapping Rules	Target: π-PEWS
Action	– An *Action* in the source model corresponding to an external **Business Collaborator** is mapped to an Operation in target model. – The **Action:name** in the source model is transformed into **Operation:name** in the target model.	Operation: alias
Service Activity	– The **ServiceActivity** in the source model is mapped to a **Composite Operation** in target model when more than one **Actions** are called. – If **Composite Operation** is generated for a given **Service Activity** then the **ServiceActivity:name** in the source model is mapped to **CompositeOperation:name** in the target model.	Type Operation, Composite Operation
Control Nodes	– The **Control Node** in the source model is mapped to a **Operator** in target model. According to the type of **Control Node** (merge, decision, join, fork) the expression of the **Composite Operation** is: • Sequence if no **ControlNode** is specified; • Parallel - Sequence for a **ControlNodes** pattern fork – join; • Choice - Sequence for a **ControlNodes** pattern decision – merge	Operator
Business Collaborator	A **BusinessCollabortor:isExternal** in the source model generates a **Namespaces** in the target model	Namespace
Rule:event	The **Rule**'s attribute event in the source model is transformed into an **Event:type** of the target model. In this case attribute is mapped to an entity with an attribute. The **Event Type** of a **Rule** in the target model is determined by the Rule type: • **Event Type** of a *Precondition* **Rule** is *ActivityPrepared*; • **Event Type** of a *Postcondition* **Rule** is *TermActivity*; • **Event Type** of a *TimeRestriction* **Rule** is *Activity*	Event Type, Event
Rule: condition	The **Rule**'s attribute condition in the source model is transformed into a **Condition:expression** in the target model. In this case, an attribute is mapped into an entity with an attribute.	Condition
Rule:action	The **Rule:action** in the source model is transformed in an **Action:act** in the target model. The attribute action is mapped to an entity with an attribute. In the target model an action is executed according to the rule condition value (true/false).	Action
Policy	– Every **Policy** associated to an element (**Business Collaborator**, **Service**, **Activity**, **Action**) in the source model becomes an **APolicy** associated to the corresponding element in the target model. – The name attribute of a **Policy** in the source model becomes an **Apolicy:name** of the target model.	APolicy
Variable	Every **Variable**, and its attributes, associated to a **Policy** in the source model becomes a **Variable** associated to an **APolicy** in the target model. The variables can be used in an **APolicy**'s Condition of the target model.	Variable
Rule:event	For a **Rule** in the source model, depending on the **Event Type**, the corresponding transformation in the target model is: **Precondition, Postcondition** or **Time Restriction Rule**	Precondition, Postcondition, Time Restriction, Rule

– We used the Eclipse Modeling Framework (EMF)[6] to implement the meta-models π-SCM and π-PEWS. Starting form these meta-models, we developed the models' plug-ins needed to support the graphical representation of the π-SCM and π-PEWS models (π-ServiceCompostion Model and π-PEWS Model plug-ins).
– We used ATL[7] for developing the mapping plug-in implementing the mappings between models (π-ServiceComposition2π-PEWS Plug-in).
– We used Acceleo[8] for implementing the code generation plug-in. We coded the pews.mt program that implements the model to text transformation for generating executable code. It takes as input a π-PEWS model implementing a specific services' composition and it generates the code to be executed by the *A-policy* based services' composition execution environment.

As shown in Figure 6, once an instance of a PEWS code is obtained starting form a particular π-services' composition model it can be executed over *A-policy* based services' composition execution environment consisting of a composition engine and a *A-policy* manager. The *A-policy* manager consists of three main components Manager, for scheduling the execution of rules, C-Evaluator and A-Executor respectively for evaluating rules' conditions and executing their actions. The *A-policy* Manager interacts with a composition engine thanks to a message communication layer (MOM).

The composition engine manages the life cycle of the composition. Once a composition instance is activated, the engine schedules the composition activities according to the composition control flow. Each activity is seen as the process where the service method call is executed. The execution of an activity has four states: prepared, started, terminated, and failure. The execution of the control flow (sequence, and/or split and join) can also be prepared, started, terminated and raise a failure.

At execution time, the evaluation of policies done by the *A-policy* manager must be synchronized with the execution of the services' composition (i.e., the execution of an activity or a control flow). Policies associated to a scope are activated when the execution of its scope starts. A *A-policy* will have to be executed only if one or several of its rules is triggered. If

[6]The EMF project is a modeling framework and code generation facility for building tools and other applications based on a structured data model.
[7]http://eclipse.org/atl/. An ATL program is basically a set of rules that define how source model elements are matched and navigated to create and initialize the elements of the target models.
[8]http://www.acceleo.org/pages/home/en

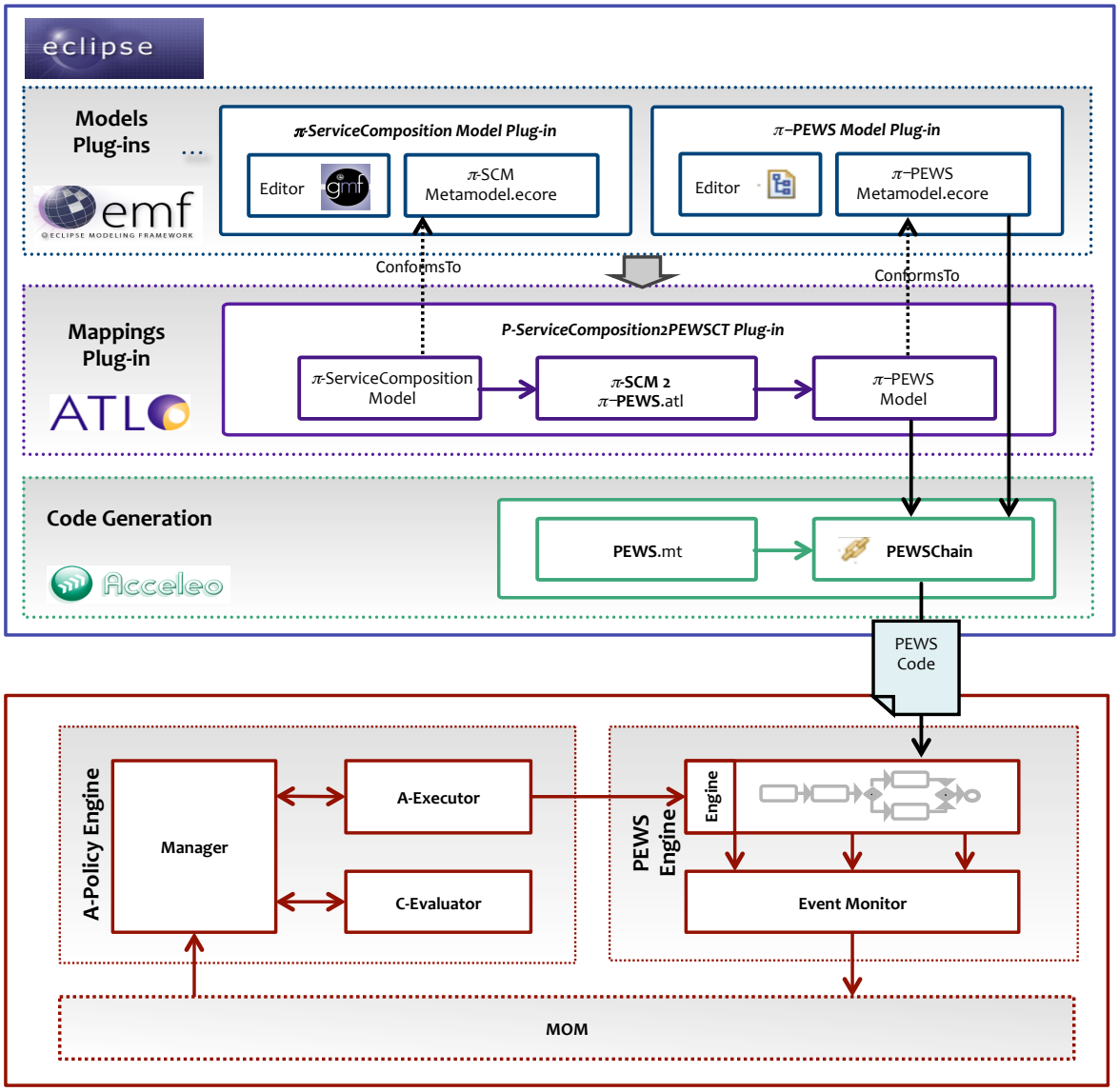


Fig. 6. πSOD-M Development Environment

several rules are triggered the *A-policy* manager first builds an execution plan that specifies the order in which such rules will be executed according to the strategies defined in the following section. If rules belonging to several policies are triggered then policies are also ordered according to an execution plan. The execution of policies is out of the scope of this paper, the interested reader can refer to [9] for further details.

VII. Related Work

Work related with our approach includes standards devoted for expressing non-functional constraints for services and services' compositions. It also includes methods and approaches for modeling non-functional constraints.

Current standards in services' composition implement functional, non-functional constraints and communication aspects by combining different languages and protocols. WSDL and SOAP among others are languages used respectively for describing services' interfaces and message exchange protocols for calling methods exported by such services. For adding a transactional behaviour to a services' composition it is necessary to implement WS-Coordination, WS-Transaction, WS-BussinessActivity and WS-AtomicTransaction.

The selection of the adequate protocols for adding a specific non-functional constraints to a services' composition (e.g., security, transactional behaviour and adaptability) is responsibility of a programmer. As a consequence, the development of an application based on a services' composition is a complex and a time-consuming process. This is opposed to the philosophy of services that aims at facilitating the integration of distributed applications. Other works, like [16] introduce a model for transactional services

composition based on an advanced transactional model. [17] proposes an approach that consists of a set of algorithms and rules to assist designers to compose transactional services. In [18] the model introduced in [19] is extended to web services for addressing atomicity.

There are few methodologies and approaches that address the explicit modeling of non functional properties for service based applications. Software process methodologies for building services based applications have been proposed in [7], [20], [21], [22], and they focus mainly on the modeling and construction process of services based business processes that represent the application logic of information systems.

Design by Contract [23] is an approach for specifying web services and verifying them through runtime checkers before they are deployed. A contract adds behavioral information to a service specification, that is, it specifies the conditions in which methods exported by a service can be called. Contracts are expressed using the language *jmlrac* [24]. The *Contract Definition Language* (CDL) [20] is a XML-based description language, for defining contracts for services. There are an associated architecture framework, design standards and a methodology, for developing applications using services. A services' based application specification is generated after several [25] B-machines refinements that describe the services and their compositions. [7] proposes a methodology based on a SOA extension. This work defines a service oriented business process development methodology with phases for business process development. The whole life-cycle is based on six phases: planning, analysis and design, construction and testing, provisioning, deployment, and execution and monitoring.

VIII. Conclusions and Future Work

This paper presented πSOD-M for specifying and designing reliable service based applications. We model and associate policies to services' based applications that represent both systems' cross-cutting aspects and use constraints stemming from the services used for implementing them. We extended the SOD-M method, particularly the π-SCM (services' composition meta-model) and π-PEWS meta-models for representing both the application logic and its associated non-functional constraints and then generating its executable code. We implemented the meta-models on the Eclipse platform and we validated the approach using a use case that uses authentication policies.

Non-functional constraints are related to business rules associated to the general "semantics" of the application and in the case of services' based applications, they also concern the use constraints imposed by the services. We are currently working on the definition of a method for explicitly expressing such properties in the early stages of the specification of services based applications. Having such business rules expressed and then translated and associated to the services' composition can help to ensure that the resulting application is compliant to the user requirements and also to the characteristics of the services it uses.

Programming non-functional properties is not an easy task, so we are defining a set of predefined *A-policy* types with the associated use rules for guiding the programmer when she associates them to a concrete application. *A-policy* type that can also serve as patterns for programming or specializing the way non-functional constraints are programmed.

Acknowledgments

This work was partially financed by the projects CLEVER, STIC-AMSUD, and MASAI. P. A. de Souza Neto was funded by CAPES/STIC-AMSUD Brazil, BEX 4112/11-3.

References

[1] M. Bell, *Service-Oriented Modeling: Service Analysis, Design, and Architecture*. Wiley, New Jersey, 2008.

[2] M. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-Oriented Computing: State of the Art and Research Challenges," *IEEE Computer*, vol. 40, no. 11, 2007.

[3] A. Watson, "A brief history of MDA," 2008.

[4] A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Ganapathy, and K. Holley, "SOMA: A method for developing service-oriented solutions," *IBM System Journal*, vol. 47, no. 3, 2008.

[5] A. W. Brown, S. K. Johnston, G. Larsen, and J. Palistrant, "SOA Development Using the IBM Rational Software Development Platform: A Practical Guide," in *Rational Software*, 2005.

[6] V. De Castro, E. Marcos, and R. Wieringa, "Towards a service-oriented MDA-based approach to the alignment of business processes with IT systems: From the business model to a web service composition model," *International Journal of Cooperative Information Systems*, vol. 18, no. 2, 2009.

[7] M. P. Papazoglou and W.-J. van den Heuvel, "Service-oriented design and development methodology," *Int. J. Web Eng. Technol.*, vol. 2, no. 4, pp. 412–442, 2006.

[8] P. Queiroz and R. Braga, "Application engineering of service-based software product lines," in *SAC*, 2012, pp. 1996–1997.

[9] J.-A. Espinosa-Oviedo, G. Vargas-Solar, J.-L. Zechinelli-Martini, and C. Collet, "Policy driven services coordination for building social networks based applications," in *In Proc. of the 8th Int. Conference on Services Computing (SCC'11), Work-in-Progress Track*. Washington, DC, USA: IEEE, July 2011.

[10] V. De Castro, E. Marcos, and J. Vara, "Applying cim-to-pim model transformations for the service-oriented development of information systems," *IInformation and Software Technology*, vol. 53, no. 19, 2011.

[11] J. Miller and J. Mukerji, "MDA guide," 2003, downloaded on 27-Jun-2014. [Online]. Available: http://www.omg.org/cgi-bin/doc?omg/03-06-01

[12] J. Gordijn and J. Akkermans, "Value based requirements engineering: Exploring innovative e-commerce idea," *Requirements Engineering Journal*, vol. 8, no. 2, 2003.

[13] J. A. Espinosa-Oviedo, G. Vargas-Solar, J.-L. Zechinelli-Martini, and C. Collet, "Non-functional properties and services coordination using contracts," in *In proceedings of the 13th Int. Database Engineering and Applications Symposium (IDEAS 09)*. Cetraro, Italy: ACM, 2009.

[14] C. Ba, M. Halfeld-Ferrari, and M. A. Musicante, "Composing web services with PEWS: A trace-theoretical approach," in *ECOWS*, 2006, pp. 65–74.

[15] P. A. Souza Neto, M. A. Musicante, G. Vargas-Solar, and J.-L. Zechinelli-Martini, "Adding contracts to a web service composition language," *LTPD – 4th Workshop on Languages and Tools for Multithreaded, Parallel and Distributed Programming*, September 2010.

[16] M.-C. Fauvet, H. Duarte, M. Dumas, and B. Benatallah, "Handling transactional properties in web service composition," in *WISE 2005: 6th International Conference on Web Information Systems Engineering*, vol. 3806. LNCS, Springer-Verlag, Octobre 2005, pp. 273–289.

[17] S. Bhiri, C. Godart, and O. Perrin, "Reliable web services composition using a transactional approach," in *e-Technology, e-Commerce and e-Service*, ser. EEE, vol. 1. IEEE, March 2005, pp. 15–21.

[18] K. Vidyasankar and G. Vossen, "A multi-level model for web service composition," in *ICWS*. IEEE Computer Society, 2004, p. 462.

[19] H. Schuldt, G. Alonso, C. Beeri, and H.-J. Schek, "Atomicity and Isolation for Transactional Processes," *ACM Transactions on Database Systems (TODS)*, vol. 27, no. 1, pp. 63–116, Mar. 2002.

[20] N. Milanovic, "Contract-based web service composition," Ph.D. dissertation, Humboldt-Universität zu Berlin, 2006.

[21] G. Feuerlicht and S. Meesathit, "Towards software development methodology for web services," in *SoMeT*, 2005, pp. 263–277.

[22] E. Ramollari, D. Dranidis, and A. J. H. Simons, "A survey of service oriented development methodologies."

[23] R. Heckel and M. Lohmann, "Towards contract-based testing of web services," in *Proceedings of the International Workshop on Test and Analysis of Component Based Systems (TACoS 2004)*, M. Pezzé, Ed., vol. 116, 2005, pp. 145–156. [Online]. Available: http://www.cs.le.ac.uk/people/rh122/papers/2005/HL05TACoS.pdf

[24] G. T. Leavens, Y. Cheon, C. Clifton, C. Ruby, and D. R. Cok, "How the design of JML accomodates both runtime assertion checking and formal verification," in *FMCO*, 2002, pp. 262–284.

[25] J.-R. Abrial, M. K. O. Lee, D. Neilson, P. N. Scharbach, and I. H. Sørensen, "The B-method," in *VDM Europe (2)*, ser. Lecture Notes in Computer Science, vol. 552. Springer, 1991, pp. 398–405.

MultiSearchBP: Entorno para búsqueda y agrupación de modelos de procesos de negocio

Hugo Ordoñez, Juan Carlos Corrales, Carlos Cobos

***Resumen*—El artículo presenta un entorno para búsqueda y agrupación de procesos de negocio denominado MultiSearchBP. Es basado en una arquitectura de tres niveles, que comprende el nivel de presentación, nivel de negocios (análisis estructural, la indización, búsqueda y agrupación) y el nivel de almacenamiento. El proceso de búsqueda se realiza en un repositorio que contiene 146 modelos de procesos de negocio (BP). Los procesos de indización y de consulta son similares a los del modelo de espacio vectorial utilizado en la recuperación de información, y el proceso de agrupación utiliza dos algoritmos de agrupación (Lingo y STC). MultiSearchBP utiliza una representación multimodal de los BP. También se presenta un proceso de evaluación experimental para considerar los juicios de ocho expertos evaluadores a partir de un conjunto de los valores de similitud obtenidos de comparaciones manuales efectuados con anterioridad sobre los modelos de BP almacenados en el repositorio. Las medidas utilizadas fueron la precisión gradual y el *recall* gradual. Los resultados muestran una precisión alta.**

***Palabras Clave*—Procesos de negocio, recuperación de información, búsqueda multimodal, agrupamiento.**

MultiSearchBP: Environment for Search and Clustering of Business Process Models

***Abstract*—This paper presents a Business Process Searching and Grouping Environment called MultiSearchBP. It is based on a three-level architecture comprising Presentation level, Business level (Structural Analysis, Indexing, Query, and Grouping) and Storage level. The search process is performed on a repository that contains 146 Business Process (BP) models. The indexing and query processes are similar to those of the vector space model used in information retrieval and the clustering process uses two clustering algorithms (Lingo and STC). MultiSearchBP uses a multimodal representation of BPs. It also presents an experimental evaluation process to consider the judgments of eight expert evaluators from a set of similarity scores obtained from previous manual comparisons made between the BP models stored in the repository. The measures used were graded precision and graded recall. The results show high accuracy.**

***Keywords*—Business processes, information retrieval, multimodal search, clustering.**

Manuscrito recibido el 18 de marzo de 2013; aceptado para la publicación el 27 de julio del 2013; versión final 16 de junio de 2014.

Hugo Ordoñez está con la Facultad de Ingeniería, Universidad de San Buenaventura, Cali, Colombia, y el Grupo de Ingeniería Telemática de la Universidad del Cauca, Colombia (correo: hugoeraso@gmail.com).

Juan-Carlos Corrales está con el Departamento de Telemática, Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca, Colombia (correo: jcorral@unicauca.edu.co).

Carlos Cobos está con el Departamento de Sistemas, Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca, Colombia (correo: ccobos@unicauca.edu.co).

I. Introducción

La apertura de los mercados y la globalización del comercio hacen que las empresas centren su atención en la oferta de nuevos productos y servicios con el propósito de atraer más clientes y de esta forma mantener o mejorar el nivel de ventas y su posicionamiento en el mercado [1]. Para lograr lo anterior, aplican estrategias que satisfacen la demanda y los requerimientos de clientes conocedores y expertos que cada día exigen más [2]. Entre estas demandas se encuentran: agilidad y calidad de servicio, rebaja de costos, disminución de tiempos, calidad de productos, agilidad en las transacciones, entre otras. Esto exige que las empresas se organicen entorno a funciones del negocio tales como: mercadeo, ventas, producción, finanzas y servicio al cliente, donde cada una de ellas se ejecutan de forma independiente según su propio modelo de negocio [3]. La aparición de los Business Process Management Systems (BPMS) permiten agilizar estas funciones dentro de la empresa facilitando su organización en torno a procesos de negocio (BP) [4], [5]. Lo anterior permite coordinar recursos humanos y tecnológicos para llevar a cabo los procesos de la empresa u organización de acuerdo con la estrategia de negocio definida.

Los lineamientos organizacionales definidos por las empresas se modelan por medio de BP, que son formados por procedimientos o actividades que colectivamente alcanzan un objetivo o política de negocio, definiendo roles y relaciones funcionales [6]. La organización por BP permite a las empresas adaptarse más eficientemente a las necesidades de los clientes, ya que los BP pueden ser modificados en cualquier momento y tantas veces como sea necesario [7].

Los BP en las organizaciones son normalmente modelados o creados por expertos, utilizando herramientas para el diseño de BP en donde plasman las operaciones o tareas que se necesita ejecutar en la organización. Las organizaciones que pretenden diseñar o modelar un nuevo BP tienen que empezar revisando grandes cantidades de información acerca de los BP existentes (normalmente almacenados en repositorios de BP).

Dentro de esta información están las instrucciones del trabajo a realizar, quién debe realizarlo y la descripción de las

conexiones con otros sistemas [8]. Esta información es almacenada en archivos que contienen los registros de transacciones conocidos como "logs" o trazas de ejecución [7], [9]. Posteriormente la información revisada sirve como base para el replanteamiento o remodelamiento de un nuevo BP que cumpla con los nuevos requerimientos de la organización [10]. El éxito en la búsqueda (descubrimiento) de los BP sobre los repositorios empresariales permite a los diseñadores reutilizar efectivamente los BP desarrollados previamente y, así disminuir el tiempo de desarrollo de los nuevos BP.

De acuerdo con lo anterior, es necesario contar con un mecanismo de gestión de información eficiente que permita buscar (descubrir) los datos generados por los BP con el propósito de encontrar aquellos BP que más similitud tienen con el comportamiento de las tareas ejecutadas en la organización y que se esperan usar para definir un nuevo BP, para un área del negocio especifica [11], [12].

En esta investigación se propone un entorno que permite el descubrimiento y agrupación de BP por medio de consultas, que contemplan características estructurales y componentes textuales. El entorno se evaluó con base en un repositorio de BP modelados con Business Process Modeling Notation (BPMN), representado en sintaxis XML, mediante el lenguaje Processing Description Language (XPDL). El entorno se basa en el modelo espacio vectorial para la representación de los BP, incorpora características de representación multimodal (que utiliza información estructural y textual) y usa algoritmos de clustering para realizar agrupaciones con base en la similitud de los BP recuperados en la consulta del diseñador.

El resto del documento está organizado de la siguiente manera. La sección 2 presenta trabajos relacionados. La sección 3 describe el entorno propuesto, sus algoritmos y algunas interfaces. La sección 4 muestra los resultados preliminares de la evaluación del modelo. Finalmente, se presentan las conclusiones y el trabajo futuro que el grupo de investigación espera desarrollar en el corto plazo

II. TRABAJOS RELACIONADOS

El tema de interés central en esta investigación es el descubrimiento de BP y la agrupación (clustering) de los mismos. A continuación se presenta un resumen de los trabajos más destacados y al final de cada sección se hace un resumen de las deficiencias de los enfoques propuestos hasta el momento.

A. Descubrimiento de BP basado en lingüística

En [11] los autores plantean un sistema de búsqueda de BP que extiende semánticamente la consulta. Cuenta con un editor de BP basado en redes de Petri e incorpora un repositorio en el cual todos los BP son etiquetados con metadatos. En este trabajo se crea un índice de búsqueda, se eliminan palabras vacías y se ponderan los términos presentes en actividades y estados del BP. El sistema cuenta con dos opciones de búsqueda, una básica y otra extendida. La búsqueda básica consulta sobre todos los modelos presentes en el repositorio o sobre un modelo en especial e incorpora WordNet como elemento de generación de sugerencias semánticas en las búsquedas. Por otra parte, la búsqueda extendida considera a cada actividad del BP como un vector de términos agregando una función de costo parcial, con la cual se calcula una función de costo total. El ordenamiento de los resultados de la consulta se realiza con los valores de la función de costo total más bajas o de menor peso.

En [13] se propone un método de compresión de lingüística basado en redes de Petri , donde se resaltan dos contribuciones realizadas, a saber: 1) un argumento teórico para establecer el grado de compresión de la lingüística, abordando la semiología (estudio de signos) de los gráficos, en donde identifican ocho variables visuales distintas que pueden ser utilizadas para codificar la información de la gráfica del BP y el color es tomado como una de las variables más eficaces para distinguir los elementos de la notación. 2) la formalización de conceptos en el modelado de flujos de trabajo (*workflows*), para lo cual toma el BP como un grafo dirigido bipartito donde P es un conjunto de nodos llamados lugares, T un conjunto de nodos llamados transiciones y Fp $(P \times T) \cup (T \times P)$ es una relación de flujo binario basado en un operador que mapea cada conjunto de nodos T. Para realizar la búsqueda del nuevo modelo ejecuta un algoritmo denominado (max-flow-min-cut) que realiza emparejamiento de nodos para encontrar el flujo máximo de coincidencias de los operadores de conexión.

En [14] se presenta un método de búsqueda basado en descomposición de BP creando un análisis híbrido entre estructura y relevancia. El algoritmo está basado en un análisis iterativo del grafo que representa al BP. La descomposición crea fragmentos de procesos reutilizables (RPF), los cuales cumplen las siguientes características: 1) Un RPF debe ser conectado de manera que todos los nodos puedan llegar desde una entrada de borde o arista, y 2) Cada RPF debe tener sólo una arista de entrada o de salida o ambos en común interconectados con otro fragmento. En este proceso se tiene como meta de búsqueda extraer la frecuencia de ocurrencia más alta en las tareas de los BP representados por los fragmentos generados.

En [15] los autores proponen un método de búsqueda de BP mediante la aplicación de reglas de asociación para información no estructurada. El proceso es llevado a cabo utilizando datos no estructurados en lugar de los registros de las aplicaciones. La ejecución del algoritmo de detección de reglas está dividida en dos: 1) la obtención de la asociación entre los documentos y procesos, 2) construcción de un modelo de lenguaje estadístico para identificación de normas relacionadas con el proceso y las actividades que se presentan en los documentos. La construcción del modelo está dividida en dos actividades principales: el algoritmo analizador, que detecta frases relacionadas con las actividades del proceso por medio de una ontología de dominio, y la identificación de patrones que utiliza una heurística, basada en los elementos de

la ontología de dominio y las sentencias del documento de búsqueda. En la recuperación de los BP se utiliza la detección de patrones, el cálculo de su frecuencia y las asociaciones de las actividades.

B. Descubrimiento de BP basado en agrupamiento (Clustering)

En [16] los autores plantean un algoritmo de clustering secuencial con el propósito de organizar una serie de objetos en un conjunto de grupos, donde cada grupo contiene objetos que son similares por un tipo de medida. Esta medida depende del tipo de objetos o datos presentes en los BP. Cada grupo está asociado con un modelo probabilístico, por lo general una cadena de Markov (al igual que el presentado en [17], [18]). Si para todos los grupos se conocen las cadenas de Markov, entonces cada secuencia de entrada es asignada a la agrupación que mejor pueda producir tal secuencia. El algoritmo desarrolla los pasos siguientes: 1) Inicializa los modelos de cluster (es decir, la cadena de Markov para cada grupo) al azar. 2) Asigna a cada secuencia de entrada el grupo que es capaz de producirlo con la mayor probabilidad. 3) La estimación de cada modelo de clúster de la serie de secuencias que pertenecen a ese grupo. Finalmente, se repiten los pasos 2 y 3 hasta encontrar los modelos de cada cluster o grupo.

En [19] plantean un enfoque de clustering que agrupa secuencias similares e identifica tópicos temáticos presentes en los BP sin la necesidad de proporcionar información de entrada. La agrupación es realizada con el propósito de encontrar información valiosa sobre el tipo de secuencias que se están ejecutando en los BP. El procedimiento de agrupación incluye: Un algoritmo alfa el cual es capaz de volver a crear el BP a través de una red de Petri, con base en las relaciones encontradas en el registro de ejecución de los BP. Métodos de inferencia que consideran el registro de ejecución como una secuencia simple de símbolos, inspirada en el modelo de Markov (al igual que el presentado en [17]) y que genera un modelo gráfico que considera cadenas de Markov de orden creciente con grafos acíclicos dirigidos. Un algoritmo de Clustering jerárquico que tiene en cuenta un amplio conjunto de trazas de ejecución de un mismo proceso, que separa las trazas en grupos y encuentra el gráfico de dependencias por separado para cada grupo. Un algoritmo genético donde las soluciones candidatas son evaluadas por una función de aptitud y cada solución es representada mediante una matriz causal, es decir, un mapa de las entradas y dependencias de salida para cada actividad.

En [18] presentan un esquema de agrupación de BP (tal como en [20], [21]) para recuperación de esquemas gráficos en grupos similares de (sub) procesos y sus relaciones. Se parte de un macro proceso para llegar hasta las actividades más sencillas, para lo cual se toma un conjunto de grafos dirigidos $G_i = <N_i, A_i>$ donde N_i es el conjunto de nodos y $A_i \subseteq N_i \times N_i$ es el conjunto de arcos posiblemente etiquetados, generando un esqueleto de agrupación típica de subestructuras. Los grafos son iterativamente analizados para descubrir en cada paso un grupo de sub-estructuras isomorfas. El clustering se utiliza para comprimir los grafos sustituyendo a cada ocurrencia de la subestructura con un nodo; este proceso se repite hasta que no haya más compresión posible.

C. Diferencias con los trabajos previos

Las propuestas anteriormente descritas en el descubrimiento lingüístico de BP se limitan al emparejamiento de entradas y/o salidas tomando como base la información textual o gráfica y las relaciones semánticas que se encuentra en la notación de estos elementos, además deja de lado el flujo de ejecución o comportamiento. En el proceso de búsqueda los resultados no tienen en cuenta similitud en patrones frecuentes, tipo de actividades, finalidad de la tarea o actividad. Por otro lado en las propuestas de descubrimiento basado en agrupación se eliminan secuencias que solo ocurren una sola vez sin tener en cuenta que pueden ser relevantes para los modelos que forman cada grupo, además la agrupación de atributos internos se mide separando su comportamiento de las propiedades estructurales y los atributos externos son medidos con datos tales como: tiempo de duración, número de errores, costo de ejecución. Esta medición de atributos hace que el costo computacional del algoritmo sea demasiado elevado.

Para alcanzar mayor relevancia de los resultados reportados en los sistemas de descubrimiento de BP, en esta propuesta se plantea un entorno que unifica en un solo espacio de búsqueda, unidades de comportamiento y características textuales de los BP, en lo que se conoce como una representación multimodal. Adicionalmente, integra el uso de algoritmos de clustering para agrupar los resultados de la búsqueda (descubrimiento) con base en la similitud de las características representadas en los modelos de BP descubiertos y lograr así una forma más efectiva de visualización de los resultados.

III. El entorno propuesto

El entorno propuesto, llamado MULTISEARCH*BP*, esta implementado sobre la tecnología Java y es soportado por una arquitectura organizada en 3 capas como se muestra en la La fig. 1. Está compuesta por: 1) un nivel de presentación desde la cual el usuario puede gestionar los BP (adicionar, eliminar, modificar y buscar BP) almacenados en el repositorio y el índice. 2) un nivel de lógica de negocio que se encarga de gestionar los BP, extraer las características estructurales y los componentes textuales de los BP e indexarlos, también responde a las opciones de búsqueda con dos tipos de respuesta: lista lineal ordenada de BP o grupos temáticos de BP que se relacionan con la consulta del usuario (diseñador) y finalmente, 3) un nivel de almacenamiento que se encarga de dar persistencia a los procesos de negocio y al índice de búsqueda. A continuación se explican cada uno de los componentes de esta arquitectura.

Formas para Adicionar / Actualizar / Eliminar: Corresponde a la interfaz grafica de usuario (GUI) usada para adicionar, modificar y eliminar BP del repositorio y del índice.

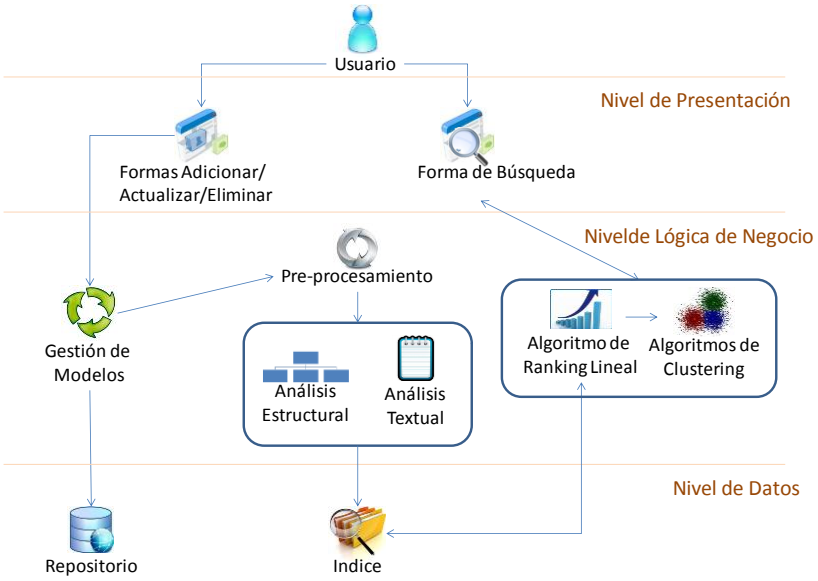


Figura 1. Arquitectura del entorno

Gestion de modelos: Este componente permite hacer gestión sobre los BP, que están en sus formatos originales XML, y representan los modelos de *BP*MN (Business Process Modeling Notation) con sintaxis XPDL (XML Process Definition Language). Estos pueden ser BP de referencia para procesos de dominio específico o BP que ejecutan un conjunto de tareas de una colección empresarial y que pueden ser reconfigurables.

Repositorio: Es la unidad central de almacenamiento y gestión, es similar a una base de datos que comparte información acerca de los artefactos de ingeniería producidos o utilizados por una empresa [10], [22]. Para la evaluación del presente entorno se usó un repositorio con 146 BP. Para cada BP se almacenan las tareas, su*b-p*rocesos y flujos de control.

Cuando la colección de BP se indexa, se realizan tres tareas fundamentales: el pre-procesamiento de cada BP, luego el analisis textual, después el analisis estructural y finalmente la creación del índice completo de la colección. Es preciso tener claro, que el índice se crea para toda la colección, pero también se puede realizar incrementalmente, es decir, uno a uno cada BP.

El **Pre-procesamiento** se encarga de convertir los términos textuales del BP a minúsculas, eliminar caracteres especiales, eliminar palabras vacías, eliminar acentos, y aplicar stemming (algoritmo de porter [27], [28]) para convertir cada uno de los componentes textuales de los BP a su raíz léxica (por ejemplo "fishing" y "fished" en "fish").

En el **Analisis textual** se lee cada uno de los elementos del conjunto $T:\{BP \mid BP\}$ presentes en el repositorio *S*, para lo cual cada uno de los elementos de *T* es representado en forma de árbol (*A*) tal que ($BP_i = A_i \rightarrow (v, x)$ donde *v* es un nodo y *x* representa las aristas). El proceso inicia tomando cada A_i, para extraer las características textuales C_{tj} (nombre de actividad, tipo actividad y descripción) para formar un vector, es decir $\{C_{tj1}, C_{tj2}, \ldots, C_{tjN}\}$, que corresponde a una fila de la matriz MC_{ij} del componente de caracteristicas textuales, donde *i* representa los BP y *j* representa las características textuales de cada uno de estos.

El **Analisis estructural** incorpora una estrategia de formación y uso de libros de códigos (codebooks) para generar unidades estructurales básicas secuenciales de los BP. Estos codebooks son construidos con base en las propiedades de similitud en patrones secuenciales frecuentes en la estructura de cada uno de los BP. Generalmente los codebooks han sido empleados en el dominio de recuperación de imágenes utilizados como histogramas de patrones visuales [29] y como vocabularios o diccionarios visuales [23], [24], [25]. Además se utilizan para analizar y buscar ocurrencias de palabras en transcripciones de texto [26].

En este paso se ejecuta el algoritmo (Parser*BP*toCodebook) que analiza la estructura de los modelos de BP almacenados en el repositorio. En este proceso se recorre de manera secuencial la estructura en árbol de los archivos XPDL donde se describe cada BP, para formar una matriz *MC* de características textuales y una matriz *MCd* de componentes estructurales (usando codebooks). Este paso se realiza tomando cada $A_i \ni vt$ (vector de transiciones), donde $vt=\{t_{j1}, t_{j2},\ldots, t_{jn}\}$, de lo cual $\forall Cd_i = (vt-1, vt); \quad i \geq 2$ con esto se tiene $A_i = \sum_{i=1}^{n} Cd_i$ formando de esta manera la matriz MCd_{ij} de componentes codebook, donde *i* representa los BP y *j* representa los codebook de cada BP. LaFigura 2 hace una representación gráfica de la manera como se forma cada uno de los codebook de un BP. De lo cual es obtenido un vector de codebooks asi: $\{\text{Start\_TaskUser}_1, \text{TaskUser\_ParallelRoute}_2, \text{ParallelRoute\_TaskService}_3, \text{ParallelRoute\_TaskService}_4\}$.

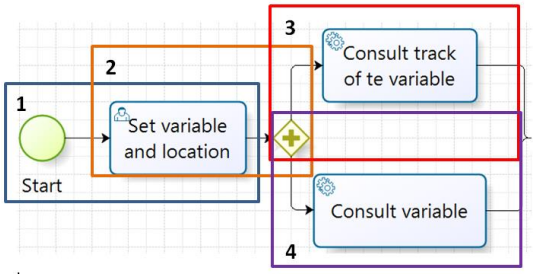


Figura 2. Estructura de cada *codebook*

	1	2	3	4	5	6	7	8	9	10	m
BP_1	1	Cd_1	Cd_2	Cd_3	Cd_4	Cd_5	Ct_1	Ct_2	Ct_3	Ct_4	Ct_m
BP_2	2	w_{ij}					w_{ij}				
BP_3	3		w_{ij}					w_{ij}			
BP_4	4			w_{ij}					w_{ij}		
BP_5	5				w_{ij}					w_{ij}	
BP_n	n					w_{ij}					w_{ij}

Figura 3. Matriz índice (MI)

El **Índice** almacena información de dos tipos: 1) Indexación de las funciones de negocios en la cual se tiene en cuenta la información textual existente en cada BP. 2) indexación estructural la cual está basada en una caracterización entre tipos de tareas, tipos de eventos y tipos de conexiones. Estas dos formas de indexación se unifican (representación multimodal) para tener una representación más exacta del objeto de estudio. El índice almacena eficientemente una estructura conceptual denominada matriz índice (*MI*) de términos por BP (similar al modelo espacio vectorial de recuperación de información [5]), que almacena en cada celda un peso (w_{ij}), el cual refleja la importancia del componente textual en su raíz léxica o codebook contra cada BP. Esta matriz se basa en la ecuación (1) propuesta por Salton [29], [27], donde $F_{i,j}$ es la frecuencia observada del componente textual o del codebook j en el BP_i. Max(F_i) es la mayor frecuencia observada en el BP_i. N es el número de BP en la colección y n_j es el número de BP en los que aparece el componente textual o codebook j. Finalmente la matriz índice $MI = \{MCd_{ij} \cup MC_{ij}\}$ puede ser resumida gráficamente como se muestra en la Figura 3. Esta figura muestra dos zonas o componentes en la *MI*, la primera, muestra el peso de los elementos de cada codebook en cada BP y el segundo el peso de los elementos textuales en cada BP.

$$w_{i,j} = \frac{F_{i,j}}{\max(F_i)} \times \log\left(\frac{N}{n_j + 1}\right) \quad (1)$$

La **Forma de búsqueda** hace referencia a un interfaz gráfica en la cual el usuario puede realizar consultas de tres formas diferentes: 1) por palabras clave (textual), 2) estructural (codebooks), y por 3) combinada de texto y estructura (es decir las dos anteriores en forma conjunta).

La consulta por palabras clave: En estas consultas el usuario puede digitar una o varias por palabras clave representadas en lenguaje natural las cuales forman un vector de consulta $qpc = \{pc_1, pc_2, \ldots, pc_n\}$. El sistema pre-procesa las palabras clave, genera un vector de consulta con los términos registrados en la *MI* y luego compara esta consulta con la parte textual del índice para entregar aquellos BP más similares a la consulta.

La consulta estructural: En esta opción el usuario tiene la posibilidad de elegir uno o varios (codebooks) de una lista de componentes estructurales formados a partir de la colección de BP existentes en el repositorio para formar el vector de consulta $qcd = \{cd_1, cd_2, \ldots, cd_n\}$. Los elementos utilizados en la consulta son comparados con la parte del índice que contiene los componentes estructurales y retorna los BP más similares a dicha consulta.

La consulta combinada de texto y estructura: Este proceso de consulta integra las dos opciones de consulta anteriores. Para realzar este proceso el sistema forma automáticamente un vector de consulta $qmg = qpc \cup qcd$, el cual se compara con cada BP registrado en la matriz *MI*, tomando las dos zonas o componentes.

Para la comparación del vector de consulta con los BP registrados en el índice se parte de los datos introducidos en la consulta, los cuales son representadas en forma de vector de términos $q = \{t_1, t_2, t_3, \ldots, t_n\}$, además se convierten todos los términos de q a minúsculas, se eliminan palabras vacías, acentos, caracteres especiales, finalmente se aplica stemming (algoritmo de porter) para convertir cada uno de los términos de q a su raíz léxica. Con la cadena de consulta procesada se ejecuta la búsqueda en el espacio elegido por el usuario, a continuación se describa cada uno de los componentes de este nivel.

Consulta: En el proceso de ejecución de la consulta el modelo ordena y filtra los BP retornados, implementando la ecuación (2) de calificación conceptual (puntuación) definida en LUCENE [28].

$$Puntuacion(q,d) = coord(q,d) \times Qnorma(q) \sum_{t \in q} \left(tf(t \in d) + idf(t)^2 \times t.getBoost() \times norm(t,d) \right) \quad (2)$$

En la ecuación anterior t es un término de la consulta q y d es el documento consultando, $tf(t \in d)$ es la frecuencia del término en el documento, definida como el número de veces que el término t aparecen en el BP d. En esta medida los documentos de mayor puntuación son los que contiene mayor frecuencia del término, $idf(t)$ es la frecuencia inversa del término t en un BP (número de BP en los que aparece el termino t), $coord(q, d)$ es un factor de puntuación basado en el número de términos de la consulta que se encuentran en el BP consultado, los BP que contienen más términos de la consulta obtienen mayor puntuación, $Qnorma(q)$ es un factor de normalización utilizado para hacer las puntuaciones (para este modelo es tomado con el valor de 1 ya que no afecta la

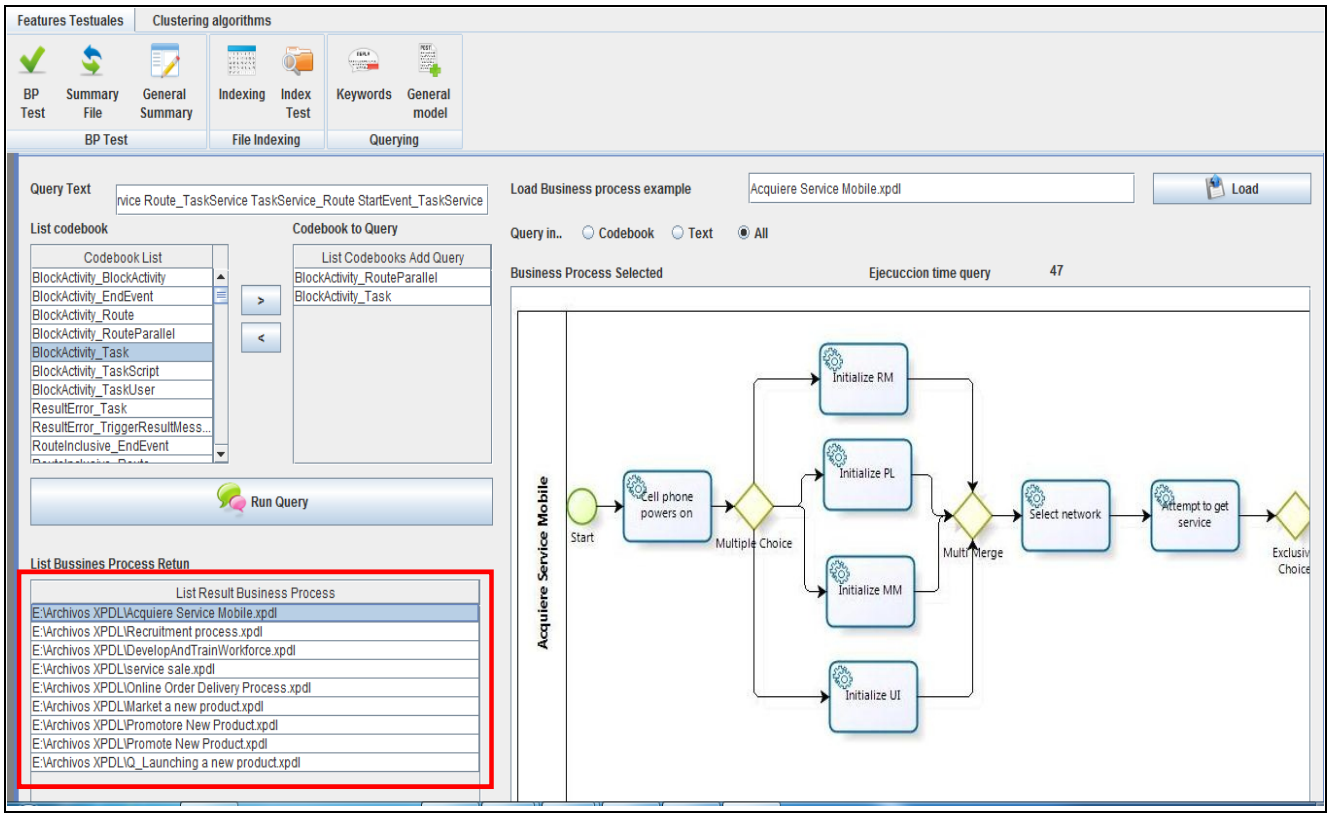


Figura 4. Opciones de consulta y despliegue de resultados en lista lineal ordenada

puntuación de cada BP evaluado). *t.getBoost*() es la ponderación del termino *t* en la consulta en este caso es igual a 1 debido que todos los términos de la consulta tienen la misma ponderación. *norm*(*t*,*d*) es un factor de ponderación en la indexación, tomado de w_{ij} en la indexación.

Una vez los resultados son ordenados y filtrados se listan en orden de acuerdo a la similitud (más similares a menos similares) que presentan con respecto a la consulta realizada por el usuario, quien puede elegir y visualizar cada uno de los modelos de BP recuperados.

Lista de resultados: Los resultados se despliegan al usuario en una lista ordenada dependiendo del nivel de relevancia, el cual es asignado obedeciendo a la puntuación definida por (2). En esta lista, el usuario puede elegir cada uno de los modelos de BP recuperados, para visualizarlos y analizarlos completamente. LaFigura 4 hace una representación gráfica de las opciones de consulta (parte izquierda central) y la lista de resultados (parte izquierda abajo enmarcada en rojo).

Nivel de agrupación: En este nivel se ejecutan los algoritmos de agrupamiento por afinidad o algoritmos de clustering [18,30] basado en las opciones de consulta explicadas en el nivel anterior, con el propósito de estructurar los resultados en grupos o familias de BP que contienen correlación en características textuales, estructurales o en ambas. Los algoritmos adaptados para este nivel son: LINGO y STC (Sufix Tree Clustering). A continuación se describen brevemente cada uno de ellos.

STC: Toma cada BP como una secuencia ordenada de términos que pueden ser textuales o estructurales, de lo cual se utiliza la información sintáctica de la secuencia para realizar la agrupación. Originalmente este algoritmo consta de tres pasos, 1) Limpiar BP, 2) Identificar clusters base y 3) Combinar clusters base. En este proyecto para aumentar el rendimiento y evitar el desarrollo de tareas redundantes del algoritmo, se eliminó el paso uno 1) Limpieza de BP, debido a que este paso se realiza previamente en el proceso de indexación.

El proceso de agrupación empieza realizando un árbol de sufijos a partir del vector que contiene todos los componentes textuales y de estructura de cada BP, se detecta una raíz, cada nodo al menos tiene dos hijos internos, las aristas entre nodos se etiquetan con una parte del texto resumen, las etiquetas de los nodos se forman uniendo el texto de las aristas, la clasificación del cluster base es realizado con la función $s(B)$, del cluster base B con frase P es: $s(B) = |B| \times f(|P|)$, donde $|B|$ = número de documentos en el cluster base B, $|P|$ = número de palabras en P que no tienen calificación 0, f = función que penaliza a las frases de una sola palabra y es lineal para frases de 2 a 7 palabras, además constante para frases mayores.

En la combinación de cluster base se tiene que en dos cluster base Bn y Bm, con tamaños $|B_m|$ y $|B_n|$. Sea $|B_m \cap B_n|$ el número de documentos comunes, La similitud entre B_n y Bm está definida como: 1 si $|B_m \cap B_n| / |B_m| > 0.5$ y $|B_m \cap B_n| / |B_n| > 0.5$ y 0 en cualquier otro caso.

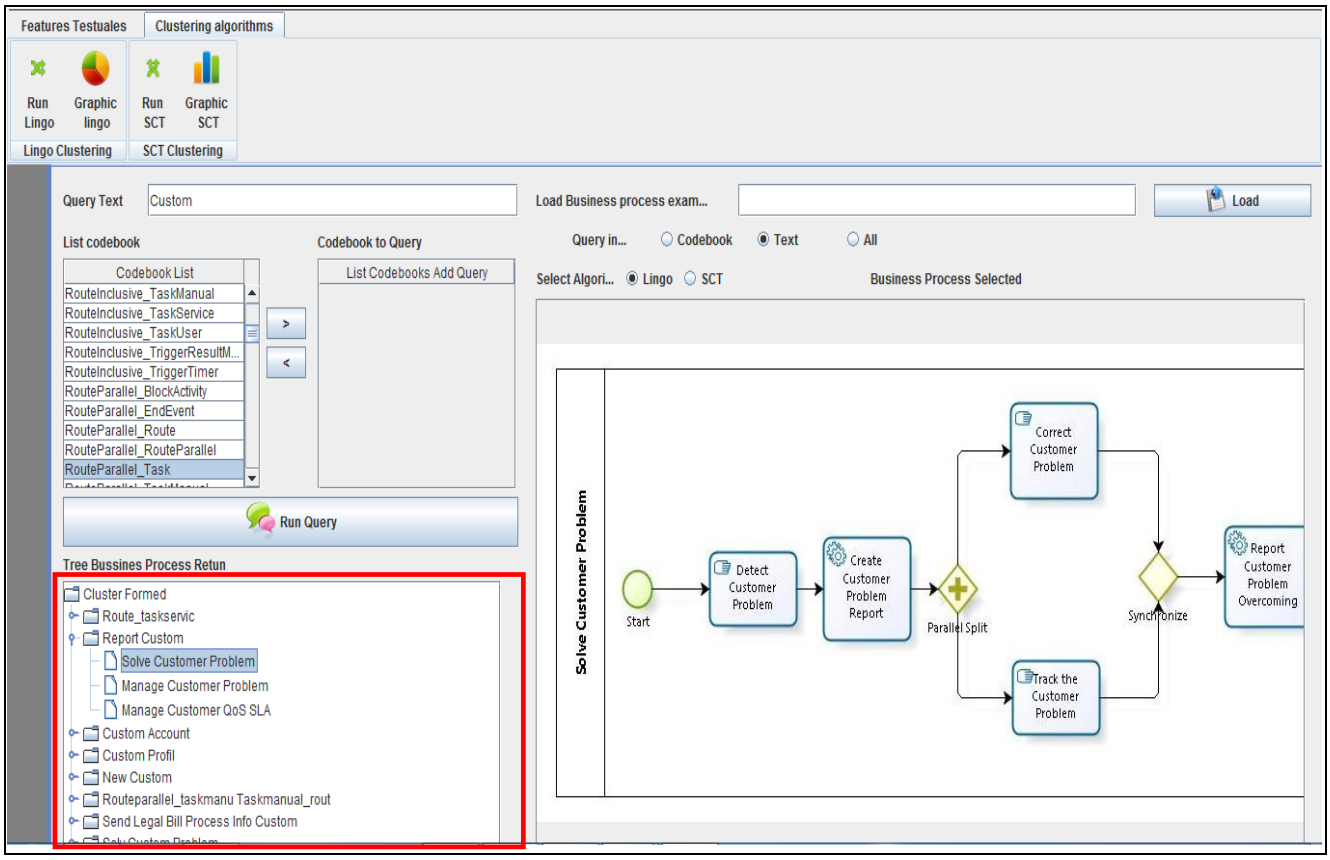


Figura 5. Opciones de consulta y despliegue de resultados en grupos temáticos

Lingo: En este algoritmo se realiza un resumen (Snippet) con los términos textuales y estructurales contenidos en cada BP descubierto en la consulta.

El algoritmo consta de cinco fases, 1) filtrado de texto, 2) extracción de características, que tiene como objeto identificar frases o términos que pueden ser candidatos para etiquetas de grupo, esto se realiza calculando el número de veces que aparecen dichas características en los BP recuperados, 3) inducción de etiquetas de cluster: en esta fase se forman descripciones significativas de grupo tomando la información de la matriz de términos por BP. Esta consta de cuatro pasos: valor del término en la matriz, el descubrimiento del concepto abstracto, la concordancia de la frase y el etiquetado, poda y evaluación, 4) descubrimiento de contenido de cada cluster: se comparan fragmentos de texto con todas y cada una de las etiquetas de grupo, para esto se forma una matriz Q en la que cada etiqueta de cluster es representada como un vector columna. De tal forma que $C=Q^TA$, donde A es el termino original de la matriz de términos por BP. De esta manera, el elemento c_{ij} de la matriz C indica el peso de adhesión del *BP j* en el grupo i, 5) formación final de clusters: se calcula con la formula valor-cluster = etiqueta-score × numero-veces, esta formación se ordena con base a la puntuación obtenida.

Al igual que en el algoritmo anterior se aumenta el rendimiento realizando la primera fase de filtrado de texto en el proceso de indexación. La Figura 5 muestra una representación gráfica de la agrupación de una consulta desplegada en forma de árbol (sección izquierda abajo enmarcada en rojo).

IV. Evaluación del entorno propuesto

Para determinar la calidad del entorno fue necesario someterlo a un proceso de evaluación experimental, con el objetivo de verificar la eficiencia en el proceso de descubrimiento de BP con base al modelo de similitud definido para las opciones de consulta que permite el entorno. Es preciso aclara que en la actualidad no se cuenta con la evaluación del proceso de agrupación. La experimentación se realizó teniendo en cuenta una colección cerrada de prueba elaborada con el juicio de ocho (8) evaluadores expertos en la temática de descubrimiento de procesos de negocio. Esta colección de prueba se realizó comparando manualmente los BP del repositorio con cada una de las consultas. En este proceso se realizaron un total de 1168 comparaciones manuales entre parejas de procesos de negocios, los cuales fueron comparados por los 8 evaluadores.

Para la evaluación se le solicitó a MultiSearchBP generar un ordenamiento (Ranking) de los 10 primeros Modelos BP (dispuestos por orden de similitud) retornados para satisfacer una necesidad definida por medio de una de las opciones de

consulta. En este sentido, es posible evaluar la calidad de los resultados obtenidos en la ejecución de esta operación del sistema, a partir de la aplicación de medidas estadísticas ampliamente empleadas en la evaluación de sistemas de recuperación de información [27], [29]. Estas medidas son la Precisión gradada (*Pg*) y el Recall gradado (*Rg*) [31], las cuales proporcionan una clasificación de los BP_t considerados similares a un BP_q de acuerdo a diferentes niveles de relevancia. De esta manera, mientras precisión y recall solo consideran la cantidad de elementos relevantes recuperados, Pg y Rg tienen en cuenta la suma total de grados de relevancia entre la consulta y los BP. En el presente trabajo se utilizaron las ecuaciones (3) y (4) [32] para evaluar *Pg* y *Rg*, relacionando el ordenamiento de los BP obtenidos por el entorno (*fe*) y el ordenamiento de las evaluaciones manuales de los expertos (*fr*). En estas ecuaciones se midió la efectividad de la recuperación de una herramienta al comparar una consulta BP_q con cada elemento de una colección BP_{ti}. Por simplicidad se considera que $BP_q = Q$ y *que* $BP_t = T$:

$$Pg = \frac{\sum_{T_i \epsilon T} \min\{f_r(Q,T_i), f_e(Q,T_i)\}}{\sum_{T_i \epsilon T} f_e(Q,T_i)}, \quad (3)$$

$$Rg = \frac{\sum_{T_i \epsilon T} \min\{f_r(Q,T_i), f_e(Q,T_i)\}}{\sum_{T_i \epsilon T} f_r(Q,T_i)}. \quad (4)$$

LaFigura 6 presenta el nivel de precisión del entorno en el descubrimiento de BP. En este proceso se desarrollaron consultas tomando como consulta 8 modelos de BP del repositorio. Los resultados de evaluación de la *Pg* en el tipo de consulta basada en la estructura alcanzaron un 41%, mientras que para las consultas realizadas por palabra clave, el entorno alcanzó un porcentaje de 76%. Finalmente las consultas realizadas con el modelo general (estructura y texto) alcanzaron el 89% de *Pg*, lo que demuestra que las consultas por modelo general (características estructurales y componentes textuales) son mucho más precisas.

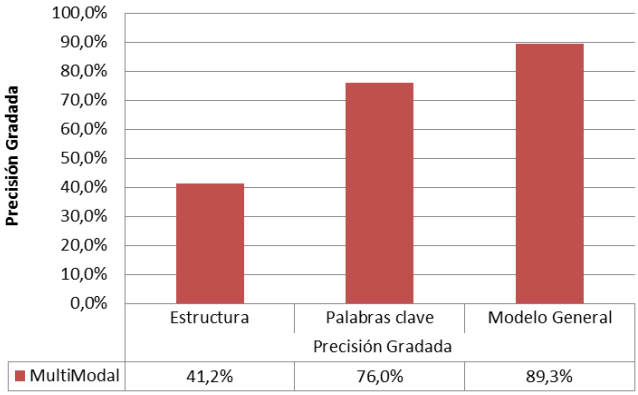


Figura 6. Grafica de precisión gradada

LaFigura 7 muestra niveles de *Rg* bajos en cada uno de los tipos de consulta, estos se encuentran en el 30% para consulta de estructura y por palabra clave (textual) mientras que el 22% para consulta por modelo general. Esto se debe a que solo se están evaluando los primeros 10 resultados y no toda la lista de resultados relevantes.

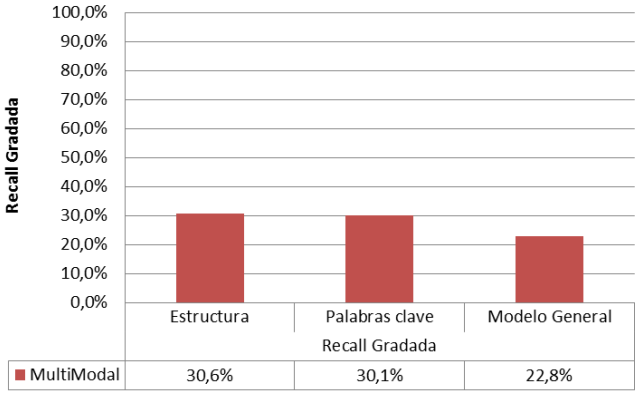


Figura 7. Grafica de recall gradada

V. Conclusiones y trabajo a futuro

En este trabajo se presentó un entorno para la búsqueda (descubrimiento) y agrupación de BP, el cual permite realizar varios tipos de consulta para ampliar el proceso de descubrimiento. Las opciones de consulta aportan flexibilidad al usuario ya que es posible replantear las búsquedas para aprovechar más el espacio de consultas y de esta forma aumentar la relevancia y pertinencia en los resultados retornados.

Los resultados obtenidos en la evaluación del entorno propuesto demuestran la eficiencia y relevancia en el proceso de descubrimiento de BP, ya que estos presentan similitud con la evaluación hecha por los expertos humanos. Alcanzando niveles de Precisión gradada que se encuentran entre el 41% como punto mínimo y 89% como punto máximo. Los resultados obtenidos en la medida de Recall gradada son bajos debido a que en el proceso de descubrimiento solo se están evaluando los primeros 10 resultados y no toda la lista de resultados relevantes, por ende no son tenidos en cuenta los BP clasificados como falsos positivos.

En el nivel de agrupación. Los grupos son formados mediante correlación y similitud directa entre características textuales, estructurales o ambas. La estructura de árbol formada permite al usuario revisar las categorías y seleccionar el grupo de mayor similitud a su consulta.

Como trabajo a futuro se propone realizar una clasificación manual de grupos de BP para poder evaluar la opción de agrupación del entorno. Evaluar la formación de grupos y comparar los resultados con otros entornos que se encuentren en el estado del arte. Incorporar ontologías de dominio específico con el propósito de realizar enriquecimiento semántico a los BP y las consultas, desarrollar un módulo de evaluación automática que genera graficas de relevancia. Ampliar la evaluación aplicando nuevas medidas para el descubrimiento de BP propuestas en [33].

AGRADECIMIENTOS

Los autores agradecen a la Universidad del Cauca y la Universidad de San Buenaventura – Cali, Colombia, por el apoyo dado al estudiante de Doctorado en Ingeniería Telemática Hugo Armando Ordóñez.

REFERENCIAS

[1] C. Cho, S. Lee, "A study on process evaluation and selection model for business process management," *Expert Systems with Applications*, vol. 38, no. 5, 2011, pp. 6339–6350

[2] Y. Gong, M. Janssen, "From policy implementation to business process management: Principles for creating flexibility and agility," *Government Information Quarterly*, vol. 29, 2012, pp. S61–S71

[3] H. Reijers, R. S. Mans, and R. van der Toorn, "Improved model management with aggregated business process models," *Data & Knowledge Engineering*, vol. 68, no. 2, 2009, pp. 221–243

[4] J. Lee, K. Sanmugarasa, M. Blumenstein, Y.-C. Loo, "Improving the reliability of a Bridge Management System (BMS) using an ANN-based Backward Prediction Model (*BP*M)," *Automation in Construction*, vol. 17, no. 6, 2008, pp. 758–772

[5] L. Xu, L. Chen, T. Chen, Y. Gao, "SOA-based precision irrigation decision support system," *Mathematical and Computer Modelling*, vol. 54, no. 3–4, 2011, pp. 944–949

[6] S. Inês, D. D. Pádua, R. Y. Inamasu, "Assessment Method of Business Process Model of EKD," vol. 15, no. 3, 2008

[7] D. Greenwood, R. Ghizzioli, "Goal-Oriented Autonomic Business Process Modelling and Execution," in: S. Ahmed and M. N. Karsiti (eds.), *Multiagent Systems*, 2009, p. 18

[8] S. Narayanan, V. Jayaraman, Y. Luo, J. M. Swaminathan, "The antecedents of process integration in business process outsourcing and its effect on firm performance," *Journal of Operations Management*, vol. 29, no. 1–2, 2011, pp. 3–16

[9] H. H. Chang, I. C. Wang, "Enterprise Information Portals in support of business process, design teams and collaborative commerce performance," *International Journal of Information Management*, vol. 31, no. 2, 2011, pp. 171–182

[10] S. Smirnov, M. Weidlich, J. Mendling, M. Weske, "Action patterns in business process model repositories," *Computers in Industry*, vol. 63, no. 2, 2012, pp. 98–111

[11] A. Koschmider, T. Hornung, A. Oberweis, "Recommendation-based editor for business process modeling," *Data & Knowledge Engineering*, vol. 70, no. 6, 2011, pp. 483–503

[12] R. Dijkman, M. Dumas, B. van Dongen, R. Käärik, J. Mendling, "Similarity of business process models: Metrics and evaluation," *Information Systems*, vol. 36, no. 2, 2011, pp. 498–516

[13] H. a. Reijers, T. Freytag, J. Mendling, A. Eckleder, "Syntax highlighting in business process models," *Decision Support Systems*, vol. 51, no. 3, 2011, pp. 339–349

[14] Z. Huang, J. Huai, X. Liu, J. Zhu, "Business Process Decomposition Based on Service Relevance Mining," *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2010, pp. 573–580

[15] D. Rosso-Pelayo, R. Trejo-Ramirez, M. Gonzalez-Mendoza, N. Hernandez-Gress, "Business Process Mining and Rules Detection for Unstructured Information," *2010 Ninth Mexican International Conference on Artificial Intelligence*, 2010, pp. 81–85

[16] D. R. Ferreira, "Applied Sequence Clustering Techniques for Process Mining," *Science*, April, 2009, pp. 492–513

[17] M. Qiao, R. Akkiraju, A. J. Rembert, "Towards Efficient Business Process Clustering and Retrieval: Combining Language Modeling and Structure Matching," *Lecture Notes in Computer Science*, vol. 6896, 2011, pp 199–214

[18] C. Diamantini, D. Potena, E. Storti, "Clustering of Process Schemas by Graph Mining Techniques (Extended Abstract)," *SEBD 2011*, 2011, p. 49

[19] D. Ferreira, M. Zacarias, M. Malheiros, P. Ferreira, "Approaching Process Mining with Sequence Clustering: Experiments and Findings," *Lecture Notes in Computer Science*, vol. 4714, 2007, pp 360–374

[20] J.-Y. Jung, J. Bae, L. Liu, "Hierarchical clustering of business process models," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 12, 2009, pp. 613–616

[21] J. Melcher, D. Seese, "Visualization and Clustering of Business Process Collections Based on Process Metric Values," *10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 2008, p. 572–575

[22] Z. Yan, R. Dijkman, P. Grefen, "Business process model repositories – Framework and survey," *Information and Software Technology*, vol. 54, no. 4, 2012, pp. 380–395

[23] H.-L. Luo, H. Wei, F.-X. Hu, "Improvements in image categorization using codebook ensembles," *Image and Vision Computing*, vol. 29, no. 11, 2011, pp. 759–773

[24] Y.-C. Hu, B.-H. Su, C.-C. Tsou, "Fast VQ codebook search algorithm for grayscale image coding," *Image and Vision Computing*, vol. 26, no. 5, 2008, pp. 657–666

[25] M. Wu, X. Peng, "Spatio-temporal context for codebook-based dynamic background subtraction," *AEU - International Journal of Electronics and Communications*, vol. 64, no. 8, 2010, pp. 739–747

[26] M. E. Fonteyn, M. Vettese, D. R. Lancaster, S. Bauer-Wu, "Developing a codebook to guide content analysis of expressive writing transcripts," *Applied nursing research: ANR*, vol. 21, no. 3, 2008, pp. 165–168

[27] C. D. Manning, P. Raghavan, H. Schütze, *An Introduction to Information Retrieval*, 2008, p. 428

[28] G. Bordogna, A. Campi, G. Psaila, S. Ronchi, "Disambiguated query suggestions and personalized content-similarity and novelty ranking of clustered results to optimize web searches," *Information Processing & Management*, vol. 48, no. 3, 2012, pp. 419–437

[29] R. Baeza-Yates, B. Ribeiro-Neto, *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999, p. 513

[30] F. Aiolli, A. Burattin, A. Sperduti, *A Metric for Clustering Business Processes Based on Alpha Algorithm Relations*, Technical report, 2011

[31] U. Küster, B. König-Ries, "On the Empirical Evaluation of Semantic Web Service Approaches: Towards Common SWS Test," *2008 IEEE International Conference on Semantic Computing*, 2008

[32] D. A. Buell, D. H. Kraft, "Performance measurement in a fuzzy retrieval environment," *SIGIR Forum*, pp. 56–62, 1981

[33] M. Becker, R. Laue, "A comparative survey of business process similarity measures," *Computers in Industry*, vol. 63, no. 2, 2012. pp. 148–167

Combining Active and Ensemble Learning for Efficient Classification of Web Documents

Steffen Schnitzer, Sebastian Schmidt, Christoph Rensing, and Bettina Harriehausen-Mühlbauer

***Abstract*—Classification of text remains a challenge. Most machine learning based approaches require many manually annotated training instances for a reasonable accuracy. In this article we present an approach that minimizes the human annotation effort by interactively incorporating human annotators into the training process via active learning of an ensemble learner. By passing only ambiguous instances to the human annotators the effort is reduced while maintaining a very good accuracy. Since the feedback is only used to train an additional classifier and not for re-training the whole ensemble, the computational complexity is kept relatively low.**

***Index Terms*—Text classification, active learning, user feedback, ensemble learning.**

I. Introduction

DURING the last decade, the Internet has become a main source of information. In November 2013 there were more than 190 million active Web sites online [1]. Since Web sites do not follow any common indexing schema, search engines are the only way to fulfill users' information needs by giving an entry-point to the Web sites with the aimed content. Besides search engines for general purposes like Google[1] or Bing[2], a number of domain specific search engines has evolved over the last years. Those search engines are tailored for the exploration of Web documents from a specific domain. Prominent domains for domain-specific search engines are hotels, restaurants, products or job offers. In contrast to general search engines, these specialized engines provide additional value based on pre-defined knowledge of their respective domains. This knowledge can be used e.g. for offering a faceted search interface, for organising the indexed Web documents or for giving recommendations based on previously viewed Web documents. Since Web documents are normally not annotated with meta information on their content, there is a need to infer this information automatically. One common method for this is the use of machine learning techniques, in particular text classification to identify appropriate class labels from the pre-defined knowledge that match the content. For the use within e.g. a hotel search engine these labels could be the focus of the hotel (*business*, *sports*, *family*, etc.) or for job search engines those could be the field of work (*IT*, *sales*, *medical*, etc.).

Traditional classification approaches require a huge number of manually labeled training instances. In static environments where domains do not adapt over time this results in a large initial effort for human annotators. In dynamic environments where the terminology changes over time, a constant annotation of large number of training instances is required which is not feasible. Hence, there is the need for an efficient solution which provides an excellent classification accuracy with less manual effort compared to traditional machine learning systems and which learns during run-time.

In this paper we present a solution which identifies Web documents that are most helpful for the system's accuracy to be annotated manually and hence to be used for the iterative improvement of the overall text classification system. The solution combines different well-known machine learning techniques such as Ensemble Learning and Active Learning but aims at having fewer time requirements compared to existing solutions.

The remainder of this paper is structured as follows. Section 2 gives an overview on fundamentals and related work in the fields relevant to our work. Based on this, our concept is presented in Section 3. Section 4 presents the methodology and the results of an extensive evaluation with 10,300 Web documents. Our achievements and future work are summarized in Section 5.

II. Fundamentals and Related Work

In this section, we give an overview on important concepts for our work. After a general introduction into the topic of *text classification* and its state-of-the-art we give insights into two general machine learning foundations for our work: *Ensemble Learning* is a technique where various machine learning results are combined into one common result. *Active Learning* allows to incorporate human feedback into a machine learning decision.

A. Text Classification

Text classification describes the automated process of assigning a text one or multiple class labels based on

Manuscript received on December 17, 2013; accepted for publication on February 6, 2014.

Steffen Schnitzer, Sebastian Schmidt, and Christoph Rensing are with Multimedia Communications Lab, Technische Universität Darmstadt, Germany (e-mail: {Steffen.Schnitzer, Sebastian.Schmidt, Christoph.Rensing}@kom.tu-darmstadt.de).

Bettina Harriehausen-Mühlbauer is with the University of Applied Sciences, Darmstadt, Germany (e-mail: Bettina.Harriehausen@h-da.de).

The first two authors contributed equally to this work.

[1]http://www.google.com

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

characteristics of the text. The class label(s) can describe various attributes of the text such as the topic or the text type. When multiple class labels can be assigned to a single text at the same time, it is referred to as multi-label classification. In our work we face a multi-label classification, but since the class labels are conditionally independent, according to [2] we break it down to a binary classification where we have to decide for each class label if it has to be assigned to a certain text or not.

In the past, a lot of work has been done in the field of text classification with various applications. As for all classification tasks, a model has to be defined first which describes instances to be classified in an abstract way. For the classification of text a widely-used model is the bag-of-words model in combination with the *term frequency-inverse document frequency (TF-IDF)* measure. The bag-of-words model represents text as an un-ordered collection of the occurring words. Since not every word has the same significance for a document, the single words are often weighted. The probably most common weighting scheme is TF-IDF, which assigns weights according to the frequency of a word within the respective text in comparison to all other texts in a corpus [3].

Using each word from a corpus as a feature where the TF-IDF values for single text instances are the feature values results in a high-dimensional space with very sparse vectors. This makes Support Vector Machines (SVMs) the most suitable classification algorithm for text classification [4].

Besides the main goal of an accurate classification, also the timing requirements for training and classification phase have been focus of research. The usage of different parallel classifiers, each of which has been trained on a sub-space of the total classes, has been presented in [5]. This approach outperforms approaches using a single classifier for the whole space of classes in terms of accuracy and speed. It is well suited for text classification tasks with hierarchical class labels but cannot be applied in settings with a large number of classes without a hierarchy.

Different methods have been presented for reducing the human effort for annotation, e.g. Fukumoto et al. present an approach that requires to have only positive examples labeled by humans [6]. More approaches are presented in Section II-C.

B. Ensemble Learning

Ensemble learning allows to combine different machine learning models into a single model. It has been shown that this improves the overall classification accuracy [7]. In this section, we will focus on the technique of *Bagging* since this proved to be most suitable for our problem. We did not employ the technique of *Stacking*, because a single most suitable classification method (SVM) has been identified. The iterative technique of *Boosting* was not used due to time performance reasons, however, the active learning part of our approach bears some characteristics of Boosting. Bagging denotes the idea to apply N instances of the same classification algorithm on N different representative random subsets of the original training set. This results in N different classifier models with different classification results [8]. The resulting labels of the single classifier can then be combined into one common result e.g. via voting or averaging, where voting is more natural for binary classifiers and averaging more natural for classifiers with numeric output. One drawback of this approach is the splitting of the complete training set, which results in smaller training sets for the single classifiers and might hence have a negative impact on the classification accuracy.

C. Active Learning

"The goal of active learning is to minimize the cost of training an accurate model by allowing the learner to choose which instances are labeled for training" [9]. The idea is to let human annotators interactively label the instances with the highest information gain and to improve the classifier by incorporating those instances into a re-training phase.

By doing so, the overall annotation effort is reduced since initially only a small number of instances needs to be annotated and afterwards only the "helpful" instances are added. This concept requires to identify the instances which can improve the classifier substantially. Strategies for selecting the most helpful instances have been focus of research for a while now [10]. Besides the advantages of this concept, also the computational effort needs to be considered. Sophisticated models like the "estimated loss reduction" [11] or the "expected error reduction" [12] are computationally very cost-intensive.

Zhu el al. [13] selects for human feedback the unlabeled instances that change their predicted class label during two consecutive learning steps or which are predicted to be in a certain class with less certainty compared to the previous step.

When making use of the previously introduced technique of Bagging, the result of voting during the classification process can be seen as a measure of certainty about the classification decision. If the single classifiers show to have differing results, the classified instance together with its label can be assumed to be helpful to improve the accuracy of the overall classifier. Li and Snoek present an approach where an ensemble of SVMs for image tag classification is re-trained iteratively with previously miss-classified examples where the correct labels are obtained via crowdsourcing [14]. The authors show that this approach leads to better classification in comparison to re-training with randomly chosen instances. The approach requires that the whole ensemble needs to be re-trained when incorporating new examples.

To conclude, we have seen that various approaches allow for a text classification which is more robust, more efficient or require less human effort. But no combination of these goals has yet been achieved.

III. Concept

A. Overview

In order to be able to use the advantages of ensemble and active learning, a combination of these two methods is sought. To achieve such a combination, two different classifiers are created which depend on each other. On the one hand, there is the ensemble learner, which employs several different classifiers using a voting scheme to find a classification decision. This base classifier is very accurate and represents the effective classification. On the other hand, there is the active learner, which is only trained with documents where the base classifier is very uncertain in its classification decision (ambiguous documents). This active learning classifier is specialized in these ambiguous documents and can be re-trained very fast. We call this combination the *Combined Ensemble and Fast Active Learner (CENFA)* [15].

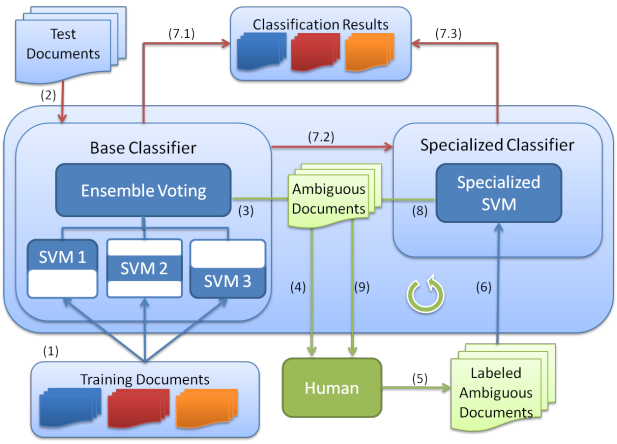


Fig. 1. The *CENFA* classifier

Figure 1 depicts the described concept. The base classifier on the left uses SVMs in a bagged ensemble. The specialized classifier on the right uses a single SVM. Ambiguous documents as identified by the base classifier are labeled by a human and used to train the specialized classifier. Test documents are then classified depending on their ambiguity either according to the results of the base classifier or the results of the specialized classifier. The re-training of the specialized classifier based on the ambiguous documents is performed iteratively.

B. Phases of Training and Classification

The training and classification using *CENFA* is described in detail along Fig. 1 in the following. The process starts with a 2-phased setup and afterwards the regular mode can be employed.

1) Setup Phase 1: At first only the base classifier is trained (1). For that, several *SVMs* are trained on different subsets of the training data acquired by bootstrapping. This application of Bagging produces a very robust initial model.

2) Setup Phase 2: In the following phase, the *CENFA* classifier is provided with classification tasks (2) and the documents to be classified are provided to the different *SVMs*. The base classifier aggregates the different results of the *SVMs* for each document and calculates a decision based on a voting scheme. Based on a confidence threshold, the base classifier decides whether the classified document appears to be ambiguous (3). All non-ambiguous classification results are discarded during this phase. By performing several classification tasks, the number of identified ambiguous documents grows. Those ambiguous documents are then annotated according to human feedback (4) which creates a certain number of labeled ambiguous documents (5). The labeled ambiguous documents are used to train a new classifier which uses a single *SVM* (6). This classifier is trained exclusively with documents that appear ambiguous to the base classifier and is therefore specialized on such documents where the base classifier shows weakness.

3) Regular Mode: Now that the specialized classifier is initially trained, the *CENFA* can enter the regular mode and be used for classification. Documents are first classified by the base classifier (2). For documents that do not appear to be ambiguous, the classification result is output directly (7.1.). When a document appears to be ambiguous to the base classifier, the classification decision is translocated to the specialized classifier (7.2). Now the specialized classifier calculates the decision based on the single *SVM* and populates it (7.3). Here the specialized classifier may identify the document as ambiguous and add it to the ambiguous documents (8). More classification tasks will make the number of ambiguous documents, which are then given to a human for feedback (9), grow again. This is used to re-train the specialized classifier and re-train it iteratively afterwards (5,6,8,9). This leads to steady improvement of the overall classifier during its usage.

This method combines the efficacy of ensemble learning, represented by the bagged base classifier and the efficacy of active learning represented by the fast iteratively trained specialized classifier. Through the simple interfaces, the inner combination of the two classifiers can be hidden and the classifier can be used as a simple active learning classifier.

IV. Evaluation

In order to prove the success of our approach we ran an extensive evaluation from which we present selected results in this section. Before presenting the results themselves, we give insight into the methodology we used for evaluation.

A. Methodology

For evaluation a corpus of 10,300 German Web documents containing job offers was used. The documents do not contain any HTML markup but the pure textual content of the Web sites. Each of these documents was annotated with one or multiple class labels which represent the job offer's respective

TABLE I
CLASSES CONSIDERED FOR EVALUATION TOGETHER WITH THE NUMBER OF POSITIVE AND NEGATIVE EXAMPLES

ID	Name	Positives	Negatives
SD	Software Development	2,077	8,223
TM	Technical Management	1,727	8,573
Sales	Sales	1,587	8,713
P-QA	Production & Quality Assurance	1,501	8,799
TDD	Technical Development & Design	1,069	9,231

field(s) of work. A set of 103 different labels was used for annotation. On average, each instance was annotated with 4.25 labels with a standard deviation of 1.86.

For the purpose of evaluation we considered only the five classes that showed to have the largest number of positive examples, which are instances annotated with the respective class label. This allows to have a large evaluation corpus available. As mentioned above, the multi-label classification problem is solved by building binary classifiers for each single label. The number of positive and negative examples for each classifier are shown in Table I. All instances not annotated with the respective class label are considered as negative training example for this class. Because the evaluation corpus is highly unbalanced, we applied a resampling of the data to achieve a better balanced data distribution. By doing so, we aim to obtain a more robust classifier.

The whole corpus used for the evaluation was preprocessed consistently so that the different classifiers where able to perform their work on the same feature set. To obtain the numeric vectors required for SVM classification, the TF-IDF statistics are gathered for the 10,000 most used words by applying a german tokenizer without using a stop word list or a stemming algorithm.

For simulating the interactive feedback given by the human annotators, we also used parts of this evaluation corpus. For each instance where the classifier decides to request the human for feedback we provide the label from the evaluation corpus. Based on this, we achieve a division of the training data into three sub-sets:

1) Subset *A* is used to train the base classifier.
2) The elements of subset *B* are classified by the base classifier and if an element is identified as ambiguous it is passed to the specialized classifier as training data together with its annotation. All elements that were identified as ambiguous form subset *S*.
3) Subset *C* is used for the evaluation (testing) of the overall CENFA classifier.

Since the goal of our work is a classification approach that can classify instances with the same accuracy as traditional ensemble learning approaches but with reduced manual human effort and with a better timing behavior, we need to compare our approach to other approaches. These approaches will be explained in the following. Subset *S* is the set of ambiguous instances which is used to train the specialized classifier of the CENFA classifier. The *Random* classifier approach uses set *R*, a random selection of elements from subset *B*, to train the specialized classifier. The number of elements in this selection is similar to the number of ambiguous instances the *CENFA* approach uses to train the specialized classifier. In other words, subset *R* is chosen to be of the same cardinality as subset *S*, while both are subsets of set *B*. The aim of this approach is to verify the suitability of using ambiguous instances for incorporating user feedback instead of training the specialized classifier with random instances. In order to examine the benefit of not retraining the base classifier with the ambiguous instances but only the specialized classifier we introduce the *Extended* classifier approach. After having recognized a number of instances as ambiguous, the whole ensemble is re-trained and the accuracy and run time of this approach are compared to the training of *CENFA*'s specialized classifier only. Last but not least, our approach is evaluated against the *Random Single SVM (RSSVM)* approach that uses a single SVM trained with the subset *A* and a random selection from set *B*. It has to be noted that *CENFA*, *Random*, *Extended* and *RSSVM* are all trained with the same amount of training data but the instances used and the overall system architecture vary across these approaches.

The three different classifiers for comparison each have a separate purpose. The *Random* delivers insights on the *accuracy* performance of *CENFA* compared to a classifier which does not use the *active learning* methodology for selecting ambiguous instances. The *RSSVM* delivers insights on *CENFA's accuracy* performance compared to a classifier which does not use the *ensemble learning* methodology and additionally the training time difference to a single *SVM* setup. The *Extended* delivers insights on the *accuracy* performance compared to a classifier which does not apply the provided compromise. Here *CENFA* was expected to be outperformed while being much faster. Table II provides an overview of the different classifiers with their used training sets and their evaluation purpose.

The *CENFA* architecture and the evaluation concept allow to tune different parameters and examine their influence on the overall accuracy in order to determine the best setting. The *dividing factor* denotes the division into the subsets *A*, *B* and *C*; in particular the given number represents the fraction of data that is assigned to subset *A*. Subsets *B* and *C* always hold the same number of instances. Hence, e.g. a *dividing factor* of 0.7 means that *A* consists of 70% of the instances from the evaluation corpus, *B* of 15% and *C* of 15%. A higher *dividing factor* results in a larger training set *A* but a smaller number of instances for the training of the specialized classifier. The second parameter which can be varied is the *confidence value*, which denotes the decision threshold of the ensemble learner up from which an instance is considered as ambiguous. If this is chosen to be very low then only a very small amount of instances from *B* are considered as ambiguous and used for the training of the specialized classifier. Further, the specialized classifier gets only a small amount of instances from *C* assigned for training since the base classifier decides

TABLE II
CLASSIFIERS WITH TRAINING SETS AND EVALUATION PURPOSE

	training sets		
Classifier	base	special	evaluation baseline with the following purpose
CENFA	A	S	proposed approach
Random	A	R	accuracy when using random instead of ambiguous instances
RSSVM	–	A+R	accuracy/timing behavior without ensemble
Extended	A+S	–	accuracy/timing behaviour with complete retraining

on the class for most of the instances. The last parameter which can be tuned is the *number of bagged SVMs* for finding a good trade-off between robustness of the ensemble and accuracy of the single classifiers.

We evaluate the approaches by calculating the *accuracy* of the classification based on a 10-fold cross-validation. Further, we examine the time required for building the classifiers using the different settings. The underlying SVM algorithm's implementation, used by all classifiers during evaluation, applies the *Sequential Minimal Optimization (SMO)* [16] algorithm with the default parameters provided by the Weka[3] framework.

B. Results

The different parameters were evaluated for their best setup before the actual evaluation results were acquired using this setup. This parameter evaluation showed that the five different classes require different tuning of the parameters to achieve the best possible results. This shows that those parameters should be evaluated and tuned differently for every scenario the *CENFA* algorithm is used in. However, to gain comparable evaluation results, the tuning parameters for the five different classes were chosen similarly. The values used for the parameters can be found below in Table III.

TABLE III
VALUES CHOSEN FOR THE TUNING PARAMETERS

Parameter	Chosen Value
Dividing Factor	0.70
Confidence Value	0.70
Number of Bagged SVMs	10

CENFA and the three different classifiers used for comparison were evaluated considering different corpus sizes. Besides the 100% corpus with 10,300 job offers, also 75%, 50%, 25% and 10% corpus sizes were used. To present the results for accuracy evaluation in a compact way, 10% and 100% corpus size were chosen for presentation in this paper only. These extreme values were chosen to show on the one hand the feasibility of the approach with a small training set only and on the other hand the increasing accuracy for a large data set. The overall trend was similar for all corpus sizes and the accuracy values were steadily increasing with increasing corpus size for all approaches presented.

[3]http://www.cs.waikato.ac.nz/ml/weka/

When using 10% of the evaluation corpus for evaluation, on average 14,5% of the instances were declared as ambiguous by the base classifier. Using the full evaluation corpus, 4.47% were declared as ambiguous. This trend is natural since the base classifier becomes more robust with a higher number of training instances.

In what follows we highlight different aspects of our evaluation.

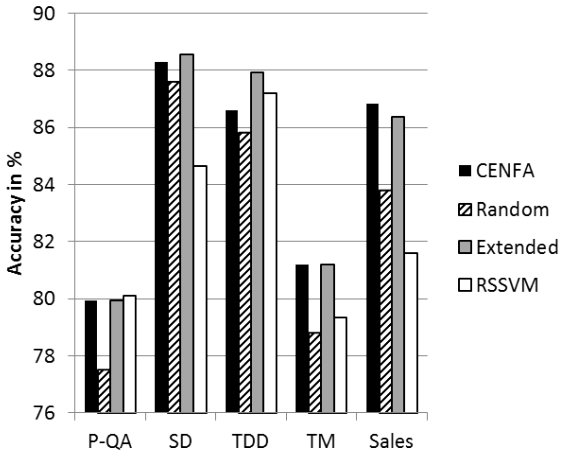


Fig. 2. The accuracy at 10% corpus size

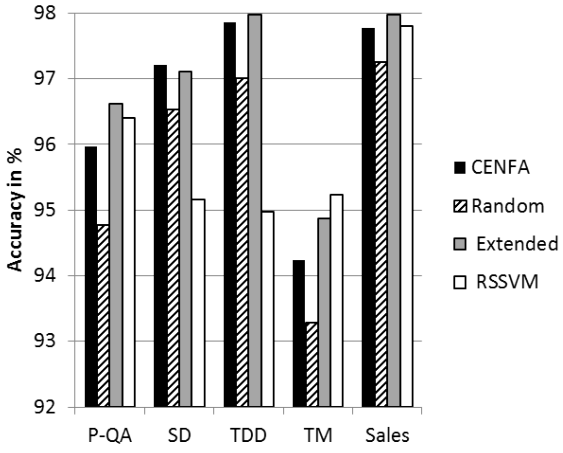
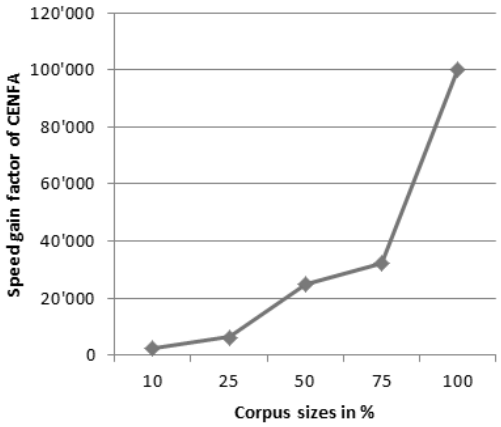
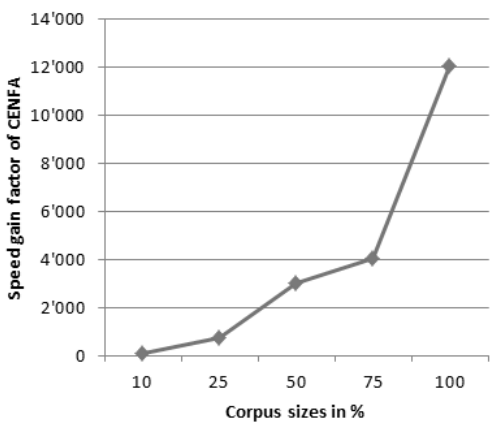


Fig. 3. The accuracy at 100% corpus size

1) Overall Accuracy: Figure 2 and Figure 3 show the performances of all the classifiers. *CENFA* can be seen to



(a) Speed gain factor of CENFA compared to RSSVM



(b) Speed gain factor of CENFA compared to Extended

Fig. 4. Compared time performance of different classifiers for different corpus sizes

perform very differently for the different classes at 10% corpus size in Figure 2. It ranges from 79.94% accuracy for the P-QA class up to 88.27% for the SD class, reaching an average of 84.57% with a standard deviation of 3.74%. This variation is likely due to the generally chosen tuning parameters which are fixed for the five classes instead of being tuned individually. Figure 3 shows that this variation decreases for bigger training corpora because the classifiers are trained thoroughly and reach towards the upper boundary of 100% accuracy. Here, *CENFA* can be seen to achieve between 94.25% and 97.86% accuracy, reaching an average of 96.61% and a standard deviation of 1.52%. Interestingly, the accuracy along the classes does not correspond with the number of positive examples per class.

2) Active Learning Accuracy: Figure 3 shows that the *CENFA* classifier was able to outperform the *Random* classifier at a corpus size of 100% in every class by between 1.2% and 0.5%. Figure 2 shows it outperforms the *Random* classifier at a corpus size of 10% by between 0.7% and 3.0%. By applying *active learning*, *CENFA* reaches higher accuracy for the same amount of training documents. By comparing Figure 2 and Figure 3 one can see that a larger number of training documents results in a better accuracy. Combining these two observations this means on the other hand that the *CENFA* classifier can reach the same accuracy by using fewer training documents compared to the *Random* classifier and therefore is more efficient.

3) Ensemble Learning Accuracy and Computational Complexity: As shown in Figure 3, the *CENFA* classifier was able to outperform the *RSSVM* classifier at a corpus size of 100% for two classes by 2.1% and 2.9% and reaches the same accuracy for one class. The *RSSVM* classifier on the other hand outperforms the *CENFA* for two classes by 0.4% and 1.0%. Figure 2 shows that *CENFA* outperforms the *RSSVM* at a corpus size of 10% for three classes by between 1.9% and 5.2% while the *RSSVM* is more accurate for two classes by 0.2% and 0.6%.

The differences of the results between the classes is due to the fixed tuning parameters which causes *CENFA's* performance to vary for the different classes while the *RSSVM* is only influenced in terms of numbers of training documents by those parameters. On average, the *CENFA* classifier reaches higher accuracy and can therefore be regarded as more effective.

However, another interesting evaluation parameter besides the accuracy is the build time. For the 100% corpus size the *CENFA* base classifier took 4,976.41 seconds to train and the special classifier took 0.05 seconds to re-train on average. The *RSSVM* took about 613.31 seconds to train. This means the *RSSVM* is about 8 times faster on the first training but *CENFA* is up to 12,000 times faster on every re-train iteration which makes it more efficient in the long run. Additionally, the speed gain factor of the *CENFA* compared to the *RSSVM* increases from small corpora to larger corpora which can be seen in Fig. 4a.

4) Uncompromising Accuracy and Computational Complexity: Figure 3 shows that the *Extended* classifier outperforms the *CENFA* classifier at a corpus size of 100% in all but the *SD* class. However, the maximal improvement of the *Extended* is at 0.6% and the average is at 0.15%. At a corpus size of 10% the *CENFA* reaches the same accuracy as the *Extended* for two classes, is outperformed for two classes by 0.3% and 1.3%, and reaches a higher accuracy for the Sales class by 0.5%. The average accuracy loss of the *CENFA* against the *Extended* is at 0.2%. That means that the *CENFA* classifier almost retains the *Extended* classifier's efficacy.

Again the build time of the classifier at a corpus size of 100% is also considered. Compared to a re-train build time of 0.05 seconds of *CENFA*, the *Extended* took 5,106.64 seconds. This means *CENFA* is up to 100,000 times faster and thus proves to be more efficient. Also, the speed gain factor of *CENFA* compared to the *Extended* increases from small corpora to larger corpora which can be seen in Figure 4b.

V. Conclusion and Future Work

The evaluation shows that the CENFA learner provides a combination of the strengths of ensemble and active learning. It is able to increase efficacy and efficiency compared to pure ensemble and active learning respectively. Compared to a standard combination of ensemble and active learning it almost retains the effectiveness and increases the efficiency substantially. In terms of time, the CENFA is up to 100.000 times faster.

The provided solution of the CENFA learner was created to classify Web documents and especially job offers. The approach needs to be evaluated in other domains of Web documents. It would also be interesting to apply this method in different classification scenarios, where entities other than Web documents have to be classified. The concept is independent from the underlying algorithm used (SVM), hence different algorithms can be tested in such an environment. Even base and specialized classifier could be applied using different algorithms. CENFA was evaluated with pre-annotated training sets simulating human feedback. Applying the algorithm in an actual active learning environment is a required step of evaluation in order to prove its suitability in a real-world scenario.

Acknowledgements

The work presented in this paper was partly funded by the German Federal Ministry of Education and Research (BMBF) under grant no. 01IS12054 and partially funded in the framework of Hessen Modell Projekte, financed with funds of LOEWE-State Offensive for the Development of Scientific and Economic Excellence (HA project no. 292/11-37). The responsibility for the contents of this publication lies with the authors. We thank kimeta GmbH for the essential help assisting with building the evaluation corpus.

References

[1] Netcraft, "November 2013 web server survey," http://news.netcraft.com/archives/2013/11/01/november-2013-web-server-survey.html, year 2013, [Online; accessed 18-November-2013].

[2] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008, vol. 1.

[3] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," *Information Processing Management*, vol. 24, no. 5, pp. 513–523, 1988.

[4] T. Joachims, "A statistical learning learning model of text classification for support vector machines," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001, pp. 128–136. [Online]. Available: http://dl.acm.org/citation.cfm?id=383974

[5] N. Tripathi, M. Oakes, and S. Wermter, "A fast subspace text categorization method using parallel classifiers," in *Computational Linguistics and Intelligent Text Processing*. Springer, 2012, pp. 132–143. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-28601-8_12

[6] F. Fukumoto, Y. Suzuki, and S. Matsuyoshi, "Text classification from positive and unlabeled data using misclassified data correction," in *Proceedings of the the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, 2013, pp. 474–478.

[7] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2011.

[8] C. C. Aggarwal, *Mining text data*. Springer, 2012.

[9] B. Settles, M. Craven, and L. Friedland, "Active learning with real annotation costs," in *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, 2008, pp. 1–10. [Online]. Available: http://dl.acm.org/citation.cfm?id=1557119

[10] Y. Fu, X. Zhu, and B. Li, "A survey on instance selection for active learning," *Knowledge and Information Systems*, vol. 35, no. 2, pp. 249–283, May 2013. [Online]. Available: http://link.springer.com/article/10.1007/s10115-012-0507-8

[11] B. Yang, J.-T. Sun, T. Wang, and Z. Chen, "Effective multi-label active learning for text classification," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09. New York, NY, USA: ACM, 2009, pp. 917–926. [Online]. Available: http://doi.acm.org/10.1145/1557019.1557119

[12] B. Settles, "Active learning literature survey," *University of Wisconsin on Active Learning, Madison*, 2010.

[13] J. Zhu and M. Ma, "Uncertainty-based active learning with instability estimation for text classification," *ACM Trans. Speech Lang. Process.*, vol. 8, no. 4, pp. 5:1–5:21, Feb. 2012. [Online]. Available: http://doi.acm.org/10.1145/2093153.2093154

[14] X. Li and C. G. Snoek, "Classifying tag relevance with relevant positive and negative examples," in *Proceedings of the 21st ACM International Conference on Multimedia*, ser. MM '13. New York, NY, USA: ACM, 2013, pp. 485–488. [Online]. Available: http://doi.acm.org/10.1145/2502081.2502129

[15] S. Schnitzer, "Effective classification of ambiguous web documents incorporating human feedback efficiently," Master's thesis, University of Applied Sciences Darmstadt, Faculty of Computer Science, Darmstadt, Germany, 2013.

[16] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods - Support Vector Learning*, B. Schoelkopf, C. Burges, and A. Smola, Eds. MIT Press, 1998. [Online]. Available: http://dl.acm.org/citation.cfm?id=299105

Una propuesta para incorporar más semántica de los modelos al código generado

Sonia Pérez Lovelle, Julio C. Cue Galindo, Alexei Hernández Perenzuela,
Andry Arredondo López, Luis R. Recio Nápoles, Frances Carnero González

***Resumen*—Actualmente hay un amplio uso del paradigma Model Driven Architecture (MDA) para la generación de código a partir de modelos, pues esto garantiza menores tiempos de desarrollo y de puesta a punto. Los modelos creados a partir de los diagramas del Lenguaje Unificado de Modelado (UML) son de amplia utilización teniendo en cuenta que se trata de un estándar y además, la gran cantidad de herramientas de modelado que existen para ello. Cada diagrama de UML es un punto de vista diferente del sistema modelado, pero cada uno de estos, tiene su sintaxis y su semántica y aporta información para el código resultante. La forma de intercambiar estos diagramas entre las diferentes herramientas es a través del uso de ficheros XMI (XML Metadata Interchange). XMI es un estándar, sin embargo, no todas las herramientas de modelado tienen las opciones de importar / exportar para este formato y las que lo hacen, no permiten la total interoperabilidad entre herramientas, debido a que usan sus propias estructuras. En este trabajo se aborda la semántica del diagrama de clases y cómo se refleja esta en el código generado por la herramienta AndroMDA, precisando los aspectos que pueden mejorarse en función de la semántica de UML, a partir de la modificación de sus cartuchos.**

***Palabras clave*—AndroMDA, diagrama de clases, MDA, semántica de UML, XMI.**

A Proposal to Incorporate More Semantics from Models into Generated Code

***Abstract*—Currently, there is a widely used paradigm called Model Driven Architecture (MDA) for code generation from models, because this ensures shorter development times. The models created from the diagrams of Unified Modeling Language (UML) are widely used, considering that it is standard and a large number of modeling tools exists for it. Each UML diagram is a different view of the modeled system, but each of them has its syntax and semantics and each of these elements provides information for the resulting code. These diagrams are exchanged between different tools using XMI files (XML Metadata Interchange). XMI is a standard; however, not all modeling tools have options to import / export to this format and they do not allow full interoperability between tools, because they use their own structures. This paper addresses the semantics of class diagram and how it is reflected in the code generated by the AndroMDA tool, specifying the aspects for improvement based on the semantics of UML through modification of their cartridges.**

***Index Terms*—AndroMDA, class diagram, MDA, UML semantics, XMI.**

Manuscrito recibido el 20 de marzo de 2013; aceptado para la publicación el 29 de julio del 2013; versión final 13 de junio 2014.

Sonia Pérez Lovelle está con la Facultad de Ingeniería Informática del Instituto Superior Politécnico José Antonio Echeverría, Cuba (correo: sperezl@ceis.cujae.edu.cu).

Julio C. Cue Galindo, Alexei Hernández Perenzuela, Andry Arredondo López y Luis R. Recio Nápoles están con el Centro de Desarrollo de Aplicaciones de Tecnologías y Sistemas, Cuba (correo: {julio.cue, alexei.hernandez, andry.arredondo, luis.recio}@datys.cu).

Frances Carnero González está con el Grupo de Electrónica para el Turismo, Cuba (correo: frances@get.mintur.cu).

I. INTRODUCCIÓN

La arquitectura dirigida por modelos (MDA, por sus siglas en Inglés), propone un proceso de desarrollo basado en la realización y transformación de modelos. MDA ha permitido minimizar los tiempos y las dificultades al desarrollar aplicaciones con alto nivel de complejidad.

AndroMDA es una de las herramientas que implementan el paradigma MDA. Su ambiente de trabajo se basa en la entrada de un modelo especificado en UML [1] (Unified Modeling Language, por sus siglas en Inglés) y exportado en formato XMI (XML Metadata Interchange, por sus siglas en Inglés), por una herramienta de modelado y genera código para las tecnologías que funcionan dentro del ámbito de la arquitectura de varias capas propuesta por ella [2].

El diagrama central en este proceso de generación de código es el diagrama de clases, en el cual la sintaxis es alterada durante el proceso de diseño, por ejemplo, cuando se sugiere que para la generación de código, en el modelado se deben describir todos los atributos pasivos de una clase con visibilidad pública, en contra de todas las buenas prácticas, pero el código resultante responde a las exigencias de un correcto diseño. Existen elementos sintácticos, como la navegabilidad en determinadas relaciones que tienen una alta carga semántica y en este caso, se dejan de generar elementos. Sin embargo, para el caso de la semántica asociada a los diferentes tipos de relaciones presentes en el diagrama, no siempre es plasmada en el código resultante, lo cual provoca que existan elementos que deben ser adicionados al código una vez obtenido y por tanto, implica dedicar tiempo a tareas que en teoría están resueltas.

En las secciones II, III y IV se trata lo concerniente a los diagramas de clases de UML, el tratamiento de las

herramientas y a la semántica del diagrama de clases, respectivamente, para en la sección IV abordar la interpretación que hace AndroMDA de la semántica del diagrama de clases de UML para poder justificar en la sección VI las modificaciones necesarias en los cartuchos de AndroMDA. Finalmente, en la sección VII se muestran algunas conclusiones y el trabajo futuro para complementar la solución actual.

II. Diagrama de clases de UML

El diagrama de clases es uno de los catorce diagramas que posee UML para la modelación de un sistema. Se trata de un diagrama estructural, que muestra cada una de las clases con sus métodos y atributos, así como sus relaciones con otras clases.

Su mayor carga sintáctica está dada en las clases, mientras que la semántica se encuentra fundamentalmente en las relaciones que se establecen entre las clases.

Los tipos de interrelaciones que se pueden establecer entre las clases son relaciones de generalización/especialización, asociación, agregación y composición.

Generalización / especialización: Una generalización es una restricción taxonómica entre dos clases. Esta restricción especializa una clase general (clase padre) en una clase más específica (clase hija). Las especializaciones y generalizaciones son dos puntos de vista del mismo concepto. Las relaciones de generalización forman una jerarquía sobre un conjunto de clases. Una jerarquía de generalización induce una relación de subconjunto en el dominio semántico de clases [3].

Asociación: Es un enlace entre instancias de los tipos asociados. Un enlace es una tupla, con un valor para cada final de la asociación, donde cada valor es una instancia del tipo del final [1].

Agregación: Modela la relación *conoce-un*. La clase que agrega o conoce, no tiene que construir a la agregada o conocida, ya que esta existe fuera de su contexto. En UML esta relación se modela a través de un rombo no relleno.

Composición: Modela la relación *tiene-un*. Está formada en sus extremos por una relación *"todo/partes"*. Cuando se construye la instancia de la clase que representa el extremo *"todo",* deben construirse también las instancias de la "parte", o sea, su constructor debe invocar al constructor de las instancias miembros para que se construyan como corresponde [4].

Existe además la Clase de Asociación, (AssociationClass). Según [1], puede verse como una clase que a la vez es una asociación, o una asociación que es también una clase. Este elemento se origina a partir de una relación que exista entre clases en la que ciertas características son propias de la relación en sí y no de alguna de las clases implicadas en la relación.

III. Herramientas de modelado UML

Las herramientas de modelado de UML existen casi desde la propia aparición de este lenguaje y han aumentado notablemente en número, existiendo en la actualidad una gran variedad de ellas, tanto propietarias, como libres y de código abierto.

Sus primeras versiones solo permitían el modelado de diagramas de UML, con formatos propios por lo que no se podía intercambiar información entre las diferentes herramientas.

Las herramientas salvan y procesan la sintaxis de los diagramas de UML, más cercana a los elementos gráficos, sin embargo, no siempre realizan validaciones de estos, lo cual tiene implicaciones negativas en la interpretación de sus significado, pero que no tienen mayores implicaciones mientras solo las herramientas se dedicaban a salvar y recuperar, pero que se convierten en claves cuando se trata de generar código.

A partir de la aparición del formato XMI como un estándar, se abren las puertas para la interoperabilidad, no solo entre herramientas de modelado, aunque esto hubiera sido ideal y algunas de estas herramientas incorporan la importación y exportación usando este nuevo formato, pero de manera general no hay un respeto total del estándar y por tanto, se pierde en gran medida la deseada interoperabilidad.

Algunos estudios realizados [5], [6] indican no solo que no existe una compatibilidad total entre las herramientas de mayor uso, sino que en algunos casos ni tan siquiera se logra importar lo que antes fue exportado por una herramienta.

Por esta razón, AndroMDA exponente del paradigma MDA, declara que su mayor "compatibilidad" o "entendimiento" es con el fichero XMI generado por la herramienta Enterprise Architect, lo cual es un reconocimiento de que estos ficheros no se ajustan a lo establecido en el estándar.

IV. Semántica del diagrama de clases de UML

A pesar de que algunos autores sostienen que UML tiene una definición semántica flexible [7], [8] o no formalmente definida [9], [10], [11], que no permite la generación de código o modelos ejecutables [12], es posible a partir de la propia sintaxis de los diagramas establecer diferencias semánticas para elementos del diagrama con diferente sintaxis como se describe a continuación de acuerdo con su significado, teniendo en cuenta que los formalismos pueden ser punto de partida para tras procesos de refinamiento, llegar a la obtención del código resultante [13].

La clase como elemento fundamental en este diagrama, da significado diferente a los atributos pasivos y activos, así como a la visibilidad de estos.

Las asociaciones entre clases permiten obtener información sobre visibilidad, navegabilidad y cardinalidad, para aquellos casos en que aparezcan descripciones en sus extremos.

Para el caso de las relaciones de generalización / especialización que representan relaciones jerárquicas de herencia, la

información semántica está asociada con la posibilidad de poder establecer si se trata de herencia solapada o disjunta por un lado y si se trata de herencia parcial o total.

Las relaciones de agregación y composición, son relaciones del tipo Todo/Parte, que sintácticamente solo se diferencian por el color del rombo, sin embargo, este tipo de rombo indica si las ocurrencias del tipo Parte pueden existir independientemente de las ocurrencias del tipo Todo y por tanto, deben ser tratadas de manera diferente.

La clase de asociación que surge de la asociación entre dos clases, que aporta información tanto en la clase como en la asociación atendiendo a la semántica o significado de ambos elementos.

V. Interpretación de AndroMDA del diagrama de clases

La herramienta AndroMDA recibe un fichero en formato XMI donde deben aparecer de manera serializada todos los elementos de un diagrama de clases, sin embargo este fichero generado por una herramienta CASE no tiene en cuenta todos los elementos y en ocasiones, las propias herramientas no son capaces de reflejar todas las posibilidades que brinda UML.

Producto de esto, en el proceso de interpretación hacia el código generado dejan de tenerse en cuenta elementos que aparecen en un diagrama de clases con un significado preciso y que por lo tanto tienen una función en el resultado final.

Aunque los principales problemas se encuentran en el código generado a partir de las relaciones, en el caso de las clases, existen dos situaciones que deben ser consideradas. Una es la clase de asociación para la cual no se genera nada relacionado con la clase, solo con la asociación y la otra tiene que ver con las superclases diferentes de la primera en una herencia múltiple, que no son tenidas en cuenta en el código resultante.

Esta última situación está dada por el hecho de que el lenguaje Java no permite este tipo de herencia, sin embargo, hay estrategias para representarla, como la propuesta en [14], en la que se hace uso de interfaces.

En el caso de las relaciones, al no darle tratamiento a la herencia múltiple, no se tienen en cuenta los atributos que se le pueden adjudicar a esta desde el punto de vista del solapamiento y si es o no total.

Las relaciones de agregación y composición son interpretadas de la misma forma, a pesar de tener un significado diferente y en consecuencia, a los efectos del código que se genera no existen diferencias y no es importante especificar uno u otro tipo de relación.

VI. Modificación de cartuchos de AndroMDA

AndroMDA no se restringe solo a generar código para las tecnologías o plataformas que provee por defecto. Debido a la arquitectura basada en cartuchos con la que se ha concebido esta herramienta, es posible modificar, crear e incluir cartuchos para generar código para cualquier plataforma [15].

Para lograr dar solución a los problemas planteados, se modificaron los cartuchos Hibernate, Java y Spring tanto para la versión 1.x de UML [16], como para la versión 2.0 [17].

Los cartuchos procesan los elementos del modelo utilizando archivos de plantillas en el lenguaje Velocity, definidas dentro del descriptor del cartucho [2]. En dichas plantillas se detalla el código que se desea generar y se agregan además los valores de determinadas variables o atributos contenidos en las clases metafachadas [15].

AndroMDA presenta como elemento principal un "motor" o "*core*", encargado de guiar el funcionamiento del proceso de generación.

El motor de la herramienta detecta los cartuchos utilizados. La información de estas dependencias se encuentra en el fichero descriptor del proyecto, ***pom.xml*** situado en la raíz del proyecto. Al leer este fichero, AndroMDA es capaz de definir cuáles serán los cartuchos a utilizar y su ubicación dentro de un repositorio establecido [15], [18].

A partir de los problemas detectados y enunciados anteriormente, se modificaron algunas plantillas en los cartuchos y se crearon otras, como se indica a continuación.

Para el tratamiento de la herencia múltiple se realizaron cambios en la plantilla *Interface.vsl* del cartucho *Java* con el objetivo de generar los atributos de las interfaces y en la plantilla *hibernate.hbm.xml.vm* del cartucho *Hibernate* para garantizar incluir estos atributos en la base de datos correspondiente.

En todos los casos, la implementación se realizó teniendo en cuenta las posibles colisiones que se pueden originar a partir de tener más de una superclase, lo cual fue resuelto de acuerdo con lo expresado en [19].

Para la implementación de la clase de asociación se modificó la plantilla ***HibernateEntity.vsl*** para generar este tipo de clase y sus relaciones y se crearon dos plantillas en el cartucho ***Hibernate***: ***HibernateAssociationClassEntity.vsl*** e ***HibernateAssociationClassEntityImpl.vsl***.

Además, para garantizar el acceso a los datos de dicha clase de asociación, se crearon, tres plantillas, en el cartucho ***Spring, SpringAssociationClassDao.vsl***, ***SpringHibernateAssociationClassDaoBase.vsl*** y ***SpringHibernateAssociationClassDaoImpl.vsl***.

Por último, para lograr la diferenciación en el código generado de las relaciones de composición y agregación, se realizaron modificaciones en la plantilla ***HibernateEntity.vsl*** del cartucho ***Hibernate.*** El código modificado hace posible que en el constructor de la clase que representa el extremo "*todo*" de la relación de composición, se cree la instancia de la clase que representa el extremo "*parte*", representando así su significado real.

VII. Conclusiones

Las herramientas de modelado de UML, de manera general, no tienen en cuenta el significado asociado a cada uno de los elementos gráficos que aparecen en los diagramas de UML y

no siguen el estándar XMI, lo cual provoca que exista una brecha semántica entre el diagrama y su serialización en el fichero XMI, lo que provoca que en el código generado por la herramienta AndroMDA no se tengan en cuenta las especificidades de cada tipo de elemento, sin embargo, ha quedado demostrado que es posible modificar o extender los cartuchos y con ello lograr incorporar mayor semántica al código generado, lo cual redunda en menor tiempo dedicado a mejorar el código generado de manera automática y por consiguiente en acortar en alguna medida el tiempo de desarrollo.

Como trabajo futuro, se trabaja en la incorporación de los diagramas de transición de estados y de actividad.

REFERENCIAS

[1] OMG. *Unified Modeling Language (UML) Specification: Superstructure version 2.0*. 5/7/2004; http://www.uml.org

[2] AndroMDA *Home Page*; http://www.andromda.org.

[3] P. Nieto Soler, "Redefinición de Asociaciones en UML: Semántica y Utilización", Tesis de Maestría, Dep. de Lenguajes y Sistemas Informáticos. Universidad Politécnica de Cataluña, España, 2008.

[4] IBM. *Rational Unified Process (RUP)*; http://www-01.ibm.com/software/awdtools/rup/

[5] A. Bordón, L. García, D. O. Hernández Darién, "ACGTool: Herramienta de soporte a la instancia de la Arquitectura de Componentes Genéricos usando UML", Trabajo de Diploma, Instituto Superior Politécnico José Antonio Echeverría, La Habana, Cuba, 2007.

[6] N. Fuentes Ramírez, "Sistema automatizado para la conversión de ficheros XMI de herramientas de modelado UML", Tesis de Maestría en Informática Aplicada. Fac. Ing. Informática. Inst. Sup. Pol. José A. Echeverría. La Habana, Cuba, 2008.

[7] M. Björdanker, C. Kobryn, "Architecting systems with UML 2.0". IEEE Software. July / August, 2003.

[8] D. Thomas, "UML – Unified or Universal Modeling Language?". *Journal of Object Technology*, vol. 2, no. 1, 2003.

[9] D. Jäger, A. Schleicher, B. Westfechtel, "Using UML for Software Process Modeling", in: O. Nierstrasz, M. Lemoine, (eds.), *ESEC/FSE'99*, *Lecture Notes in Computer Science*, vol. 1687, 1999

[10] R. Heckel, J. Kuster, G. Taentzer, "Towards Automatic Translation of UML Models into Semantic Domains", *APPLIGRAPH Workshop on Applied Graph Transformation (AGT 2002)*, Grenoble, France, 2002

[11] R. B. France, S. Ghosh, T. Dinh-Trong, "Model-Driven Development Using UML 2.0: Promises and Pitfalls", IEEE Computer Society, 2006

[12] J. Hogg, "Brass Bubbles: An overview of UML 2.0 (and MDA)", *Fourth Workshop on UML for Enterprise Applications: Delivering the Promise of MDA*, IBM Software Group, OMG, Junio 2003.

[13] M. Egea González, "Una semántica formal ejecutable para OCL con aplicaciones al análisis y a la validación de modelos", Ph.D. tesis, Dept. Sist. Inf. y Computación. Univ. Complutense de Madrid, 2008

[14] S. Holzner, "La Biblia de Java 2 [Multimedia]", Madrid: Anaya Multimedia, 2000

[15] D. Pagés Chacón, "Cartucho de AndroMDA para JSF Interpretando Nueva Estrategia de Modelado", Tesis de Maestría en Informática Aplicada. Fac. Ing. Informática. Inst. Sup. Pol. José A. Echeverría, La Habana, Cuba, noviembre 2010

[16] A. Arredondo López, L. R. Recio Nápoles, "Modificación de cartuchos de AndroMDA para incluir más semántica del diagrama de clases de UML 1.4", Trabajo de Diploma, Instituto Superior Politécnico José Antonio Echeverría, La Habana, junio 2012

[17] J. C. Cue Galindo, A. Hernández Perenzuela, "Modificación de cartuchos de AndroMDA para incluir más semántica del diagrama de clases de UML 2.0". Trabajo de Diploma, Instituto Superior Politécnico José Antonio Echeverría, La Habana, junio 2012

[18] E. Hernández Lee, "Extensión al cartucho bpm4struts de AndroMDA para la generación de componentes de prueba para entornos especializados en struts 1.x". Tesis de Maestría en Informática Aplicada, Fac. Ing. Informática, Inst. Sup. Pol. José Antonio Echeverría, La Habana, Cuba, julio 2010

[19] J. Cácerez Tello, "Curso de Java, Cápsula Formativa. Los interfaces y la herencia múltiple". 2011

Comparison of Different Graph Distance Metrics for Semantic Text Based Classification

Nibaran Das, Swarnendu Ghosh, Teresa Gonçalves, and Paulo Quaresma

***Abstract*—Nowadays semantic information of text is used largely for text classification task instead of bag-of-words approaches. This is due to having some limitations of bag of word approaches to represent text appropriately for certain kind of documents. On the other hand, semantic information can be represented through feature vectors or graphs. Among them, graph is normally better than traditional feature vector due to its powerful data structure. However, very few methodologies exist in the literature for semantic representation of graph. Error tolerant graph matching techniques such as graph similarity measures can be utilised for text classification. However, the techniques like Maximum Common Subgraph (mcs) and Minimum Common Supergraph (MCS) for graph similarity measures are computationally NP-hard problem. In the present paper summarized texts are used during extraction of semantic information to make it computationally faster. The semantic information of texts are represented through the discourse representation structures and later transformed into graphs. Five different graph distance measures based on Maximum Common Subgraph (mcs) and Minimum Common Supergraph (MCS) are used with k-NN classifier to evaluate text classification task. The text documents are taken from Reuters21578 text database distributed over 20 classes. Ten documents of each class for both training and testing purpose are used in the present work. From the results, it has been observed that the techniques have more or less equivalent potential to do text classification and as good as traditional bag-of-words approaches.**

***Index Terms*—Graph distance metrics, maximal common subgraph, minimum common supergraphs, semantic information, text classification.**

I. Introduction

The research on automatic text classification task [1], [2] is one of the interesting area to the Natural Language Processing (NLP) researchers for the last few decades due to having its huge applications. The task becomes still more challenging with the ever increasing volume of complex text information especially through web-based services. State of the art approaches typically represent documents as vectors (bag-of-words) and use a machine learning algorithm, such as k-NN, Naïve Bayes, SVM to create a model and to classify new documents. But these approaches fail to represent the semantic content of the documents which is necessary for certain kind of tasks such as opinion mining, sentiment analysis etc. Therefore, in spite of being able to obtain good results, these approaches are utilized only for limited number of tasks. To overcome the limitations, the researchers are aiming to evaluate and use more complex knowledge representation structures [3], [4].

In this paper, a new approach which integrates a deep linguistic analysis of the documents with graph-based representation has been proposed for the text classification. Discourse representation structures (DRS) [5] are used to represent the semantic content of the texts and are transformed by our system into graph structures. Then, we proposed, applied, and evaluated several graph distance metrics [6] on 20 document classes from Reuters21578 text database taking 10 docs of each class for both training and testing purpose using a k-NN classifier. Later we compared the obtained results with the result obtained by traditional bag-of-words approaches.

The paper is organized as follows. Section 2 briefly describes the theoretical background related to our approach: discourse representation theory, graph representation, k-NN classifiers, and graph metrics. Section 3 presents our system and its modules. Section 4 exposes the performed experiments and discusses the obtained results. In the final section of the paper, conclusions and future work are presented.

II. Theory and Algorithms

A. Brief description of DRS

Extracting information from documents can be carried out in many ways, starting from statistic or probabilistic models to the ones involving deep linguistic structures. Our main goal in this work is to develop a technique which analyses documents in lexical, syntactic as well as semantic level.

Discourse Representation Theory (DRT), proposed by Kamp and Reyle [5] is one of the most advanced form of representing semantic context of a document. In DRT, a sequence of sentences $S_1, S_2, \ldots, S_n$ is passed into an algorithm. It starts with syntactic analysis of the first sentence S_1 and transforms it roughly top down, left to right fashion according to some DRS construction rules. This new DRS K_1 serves as a context for analyzing S_2 which in turn generates $K_{1,2}$ by appending the new semantic content to K_1.

A complete DRS expression is composed of: (a) a set of referents, which are the entities that have been introduced into

Manuscript received on January 4, 2014; accepted for publication on February 6, 2014.

Nibaran Das (corresponding author) is with the Computer Science and Engineering Department, Jadavpur University, Kolkata-700032, India (phone: +91 332 414 6766; fax: +91 332 414 6766; e-mail: nibaran@ieee.org).

Swarnendu Ghosh is with the Computer Science and Engineering Department, Jadavpur University, Kolkata-700032, India.

Teresa Gonçalves is with the Dept. of Computer Science, School of S&T, University of Évora, Évora, Portugal.

Paulo Quaresma is with the Dept. of Computer Science, School of S&T, University of Évora, Évora, Portugal, and with with L2F – Spoken Language Systems Laboratory, INESC-ID, Lisbon, Portugal.

the context, (b) a set of conditions, which are the relations that exist between the referents.

DRT provides a very logical platform for the representation of semantic structures of sentences including complex predicates like implications, propositions and negations, etc. It is also able to separately localize almost every kind of events and find out their agents and patients.

Here is an example of a DRS representation of the sentence "He drinks water.". Here, *x1*, *x2*, and *x3* are the referents and *male(x1), water(x2), drink(x3), event(x3), agent(x3, x1), patient(x3, x2)* are the conditions

[x1, x2, x3:
male(x1), water(x2), drink(x3),
event(x3), agent(x3, x1), patient(x3, x2)]

B. Brief description of GML

Graph Modeling Language (GML) [7] is a simple and efficient format for representing weighted directed graphs. A GML file is primarily a 7-bit ASCII file. Its simple format allows us to read, parse, and write without much hassle. Moreover, several open source software systems are available for viewing and editing GML files.

Graphs are represented using several keys like "graph", "node", "edge" etc. while nodes have "id" associated with them which are later referenced from the "source" and "target" attributes. Edge weights are represented through "label" attribute associated with an edge key.

C. k-NN classifier

The k-nearest-neighbour is one among the most simple and popular machine learning algorithms. These kinds of classifiers depend solely on the class labels of the training examples that are similar to the test example instead of building explicit class representation. Distance measures such as Euclidean distance, Manhattan distance are generally used to compare the similarity between two examples. In standard k-NN algorithm the majority vote of its neighbours are used to classify a new example. Usually, the number of neighbours (value of k) is determined empirically to obtain best results.

D. Distance metrics for graphs

As we have mentioned before, the goal of our current work is to make a comparative analysis of different kinds of distance metrics for text classification task.

We have taken five different distance metrics from [6], which are used in this work. They are popularly used in object recognition task, but for text categorization they have not been used popularly. For two graph $G_1$and G_2, if $d(G_1, G_2)$ is the dissimilarity/similarity measure, then $d(G_1, G_2)$ would be a distance, if d has the following properties:

(i) $d(G_1, G_2) = 0$ iff $G_1 = G_2$

(ii) $d(G_1, G_2) = d(G_2, G_1)$

(iii) $d(G_1, G_2) + d(G_2, G_3) \geq d(G_1, G_3)$

The measures that are involved in the current work follow the above rules. The corresponding distance metrics for these measures are:

$$d_{mcs}(G_1, G_2) = 1 - \frac{|mcs(G_1, G_2)|}{max(|G_1|, |G_2|)} \quad (1)$$

$$d_{wgu}(G_1, G_2) = 1 - \frac{|mcs(G_1, G_2)|}{|G_1| + |G_2| - |mcs(G_1, G_2)|} \quad (2)$$

$$d_{ugu}(G_1, G_2) = |G_1| + |G_2| - 2|mcs(G_1, G_2)| \quad (3)$$

$$d_{MMCS}(G_1, G_2) = |MCS(G_1, G_2)| - |mcs(G_1, G_2)| \quad (4)$$

$$d_{MMCSN}(G_1, G_2) = 1 - \frac{|mcs(G_1, G_2)|}{|MCS(G_1, G_2)|} \quad (5)$$

In the above equations, $mcs(G_1, G_2)$ and $MCS(G_1, G_2)$ denote maximal common subgraph and minimum common super graphs of two graphs $G_1$and G_2. Theoretically $mcs(G_1, G_2)$ is the largest graph in terms of no. of edges which is isomorphic to a subgraph of both $G_1$and G_2. The $mcs(G_1, G_2)$ has been formally defined in the work of Bunk et al. [8].

As stated earlier, finding the maximum common subgraph is a NP complete problem and, the algorithm of finding the *mcs*() is actually a brute force method, which first finds all the subgraphs of both the graphs and select the graph of maximum size which is common to both G_1 and G_2 . To increase computational speed of the program, it is modified to an approximate version of actual $mcs(G_1, G_2)$ residing on the fact that the nodes that possess a greater similarity in their local neighborhood of the two graphs have a larger probability of inclusion in the *mcs*. The two stage approach used in the present work to form the approximate $mcs(G_1, G_2)$ is as follows:

1. All the node pairs (one from each graph) are sorted according to the decreasing order of their similarity of local structures. In the present case, the number of self-loops which have equal labels in both the graphs is used for similarity measures.
2. Build the *mcs* by first adding each self-loop vertex pair (starting with the one with the highest no. of matching labels) and considering it as an equivalent vertex, then include the rest of the edges (non-self-loop edges) which satisfy the chosen self-loops in both the graphs.

In this way it can be ensured that the approximation version possesses most of the properties of a *mcs*, while complexity is contained within a polynomial upper bound.

The minimum common supergraph (MCS) [4] is formed using the union of two graphs, i.e. $MCS(G_1, G_2) = G_1 \cup G_2$. The distance metrics of Equations 1, 2, and 5 were used without modification, but those of Equations 3–4 were divided by $(|G_1| + |G_2|)$ and $|MCS(G_1, G_2) + mcs(G_1, G_2)|$, respectively to make them normalized, keeping the value of distance metrics in the range $[0, 1]$.

E. Tools

In order to extract DRS from summarized texts we used "C&C" and "Boxer" [10] [11], which are very popular open source tools available for download at http://svn.ask.it.usyd.edu.au/trac/candc. The tools consist of a combinatory categorical grammar (CCG) [9] parser and outputs the semantic

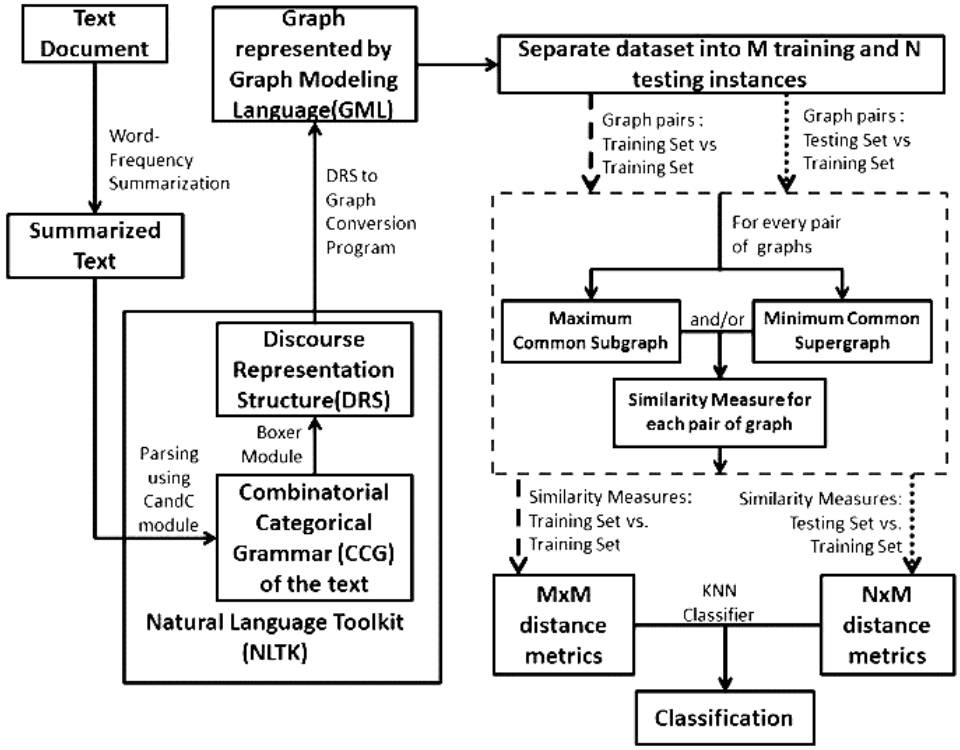


Fig 1. Block Diagram of the system showing major stages like Semantic Information Extraction, Formation of Graphs, Calculation of Distance Metrics, and Classification using k-NN Classifier

representations of texts using discourse representation structures (DRS) which are defined in Discourse Representation Theory (DRT) [5].

III. Methods

Our method involves three primary phases. The first involves extraction of semantic information from summarized documents. The second phase indulges into the conversion of DRS into a graphical structure. Finally the third one focuses on the learning phase where Distance metrics are computed on the basis of graphical structures which are further used for classification using k-NN classifier [2]. The flowchart of the entire system is shown in Fig. 1.

A. Extraction of Semantic Information

Bag-of-Words approach has been one of the most common approaches for text classification. Though the incredible amount of success achieved by this approach yet it fails to actually understand a language. A language is not merely a collection of words placed randomly. A language is defined by its grammar which binds different entities with relations that gives a document a sense of entirety with respect to its contents. Hence we have to move on from bag-of-words approaches to truly understand a language. Hence it is very essential to explore the semantic level analysis of the languages and DRT is such a framework.

However, before using DRS we need to convert it to a more dynamic data structure. We have decided to use graphs as they possess an intrinsic property that makes them suitable to represent DRS. Referents and conditions are easily represented through the nodes and edges of graph. Graphs also ensure faster traversal through semantic networks. Moreover numerous graph similarity metrics exist which can be used to compare two documents and find their similarity. Hence, a robust system may be built which can minutely observe and analyze complex semantics of natural language and efficiently categorize them.

However, we should note that the traditional $mcs()$ and $MCS()$ is a NP complete problem. To minimize the complexity, summarizations of documents are performed. Summarization is done on the basis of frequency of words. The sentences are chosen whose words occur with greatest frequency over a particular class. Throughout this process stop-words are ignored completely. The sentences are ranked in order to be able to easily choose the best ones for summarization. The summarization is done using the tool available from git://github.com/amsqr/NaiveSumm.gitmaster.

The summarized text is then sent to the C&C parser [9] to identify the CCG derivations, POS tags, lemmas and named entity tags which are then used by Boxer [9] to produce the DRSs based on the inherent semantic interpretation of the sentence.

B. Formation of Graphs

The DRS output provided by Boxer is converted to graph structure. For building the graph we used the format of Graph Modeling Language (GML). As mentioned earlier, Boxer is capable of representing various kinds of complex predicates like proposition, implication, negation etc. However, the entire

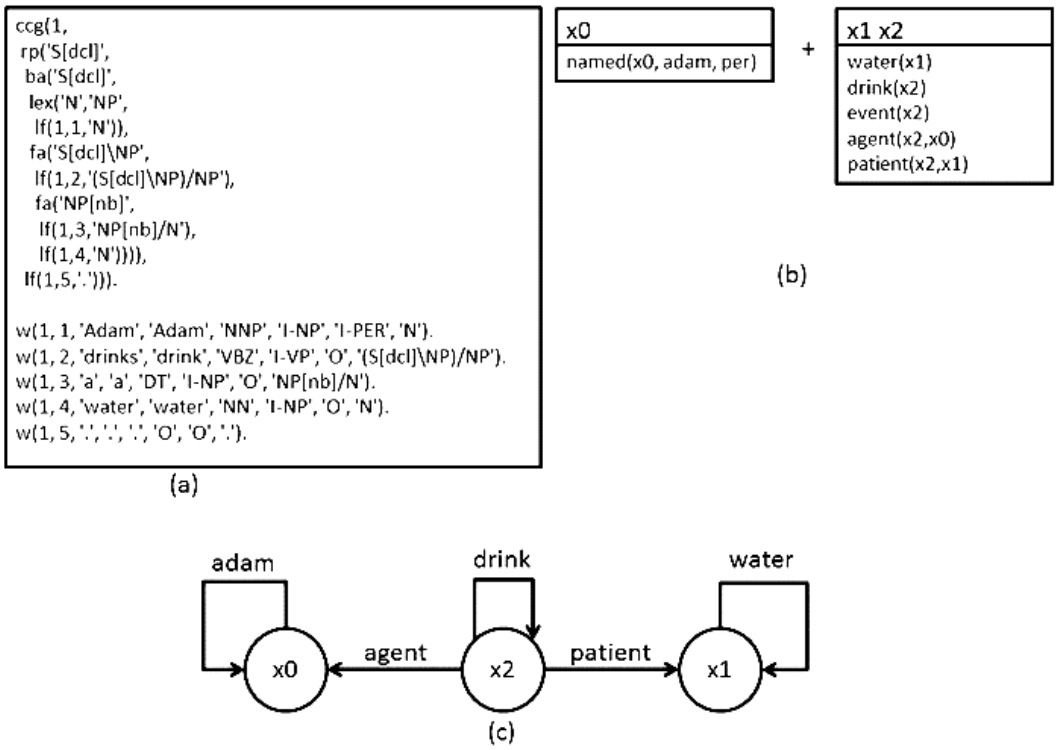


Fig 2. The transformations for the sentence "Adam drinks water": (a) C&C output, (b) Boxer output and (c) the corresponding graph

DRS structure can be broadly broken into referents or entities and conditions or relations. In the graph referents are treated as nodes and the conditions as edges. While assigning referents to vertices all equality cases are resolved beforehand. Conditions are represented as directed edges. The direction assigned is from the first referent to the second referent. In case of conditions with single referents like *male(x1)*, a self-loop is added at the vertex. Special conditions like propositions, implications are handled as conditions in the DRS and hence represented as edges between the concerned referents. Condition names are used as labels for the edges. Agent and patient are also treated as conditions of discourse, hence represented by the edge values of two referents. An example of a sentence and its transformations (syntactic, semantic and graph representation) is shown in the Fig. 2.

To measure the distance between two graphs, the approximate $mcs(G_1, G_2)$ is constructed based on the steps described in Section 2.4, it is then for the creation of $MCS(G_1, G_2) = G_1 + G_2 - mcs(G_1, G_2)$ to make it computationally faster. Fig. 3 shows the mcs and MCS of two graph sentences.

C. Classification using the different distance metrics and the k-NN classifier

It has already been mentioned that the different distance metrics (see Equations 1-5) are calculated based on the $mcs()$ and $MCS()$. The values of $mcs()$ and $MCS()$ are represented by the number of similar vertices or the number of similar edges. Thus, ten different distances are calculated based on Equations 1-5.

During the classification phase two matrices are generated for each of the above ten distance metrics. The training set is an $M \times M$ matrix formed by pairing each training data with all other training data and calculating their distance values. The testing set is a $N \times M$ matrix formed by pairing each testing data with all training data. The feature vector is hence represented as an M dimensional vector which comprises of similarity scores for each of the M training documents. The results obtained were used to evaluate the performance of each distance on the dataset (shown in Table 1 and 2).

IV. Experimental Results and Discussion

Reuters-21578 is one of the most popular text corpora that have been used for text classification. We have used a subset of this dataset. The selected dataset comprises of 20 documents (10 training and 10 testing) belonging to each of 20 selected classes, viz. *acq*, *alum*, *barley*, *bop*, *carcass*, *cocoa*, *coffee*, *copper*, *corn*, *cotton*, *cpi*, *crude*, *dlr*, *earn*, *fuel*, *gas*, *gold*, *grain*, *interest* and *ipi*. As shown in Fig 1, a summarization technique based on word frequencies is used to generate two and three sentences summarization of the entire text.

The summarized texts are then passed into the NLTK toolkit [11] where semantic information is extracted by Boxer. Then an algorithm converts DRS to graph using the format of graph modeling language. Then five different distance metrics (see Equations 1–5) are calculated on pairs of graphs, which is later used for classification using *k*-NN classifiers. The accuracies observed for the test dataset for 3, 5 and 7 nearest neighbours (*k* value) are shown in Table 1 and 2 along with a

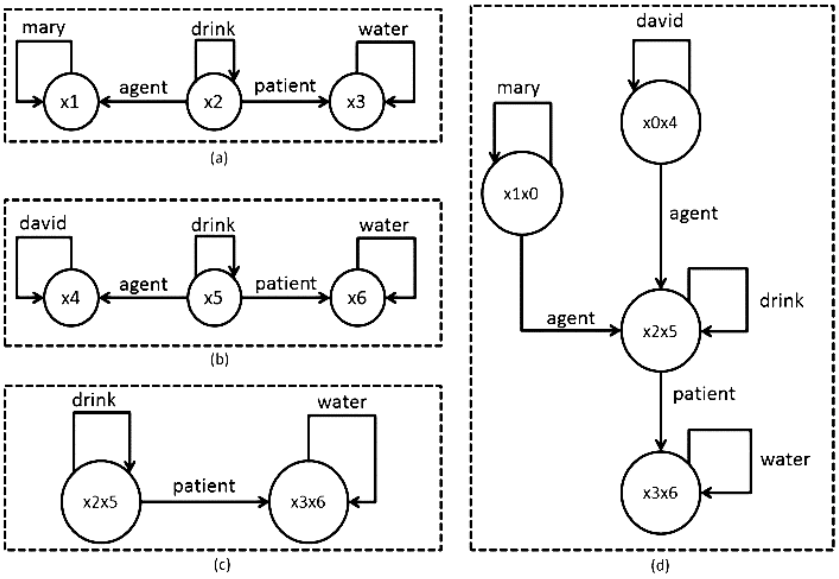


Fig 3. Graphical overview of mcs and MCS: (a), (b) graph representation of sentences meaning "Mary drinks water" and "David drinks water" (c) maximum common subgraph (d) minimum common supergraph

result for the traditional bag-of-words approach.

From Tables 1 and 2 and Fig. 4 it could be observed that all the edge based distance metrics perform better than their vertex equivalent.

Therefore, the DRS conditions or relations, which are represented by edge values, play an important role in the classification job. Since the edge values are the main indicators of the underlying semantics it can be concluded that semantic information is essential for text categorization. From Table 1 and 2 it can also be observed that average recognition accuracies of two sentences are lower than that of the three sentence summarization techniques. This can be easily visualized in Fig. 5.

The maximum accuracy observed in the present work is 51.50% for edge based experiment is for Equation 5 with 3 sentence summarization for k = 7. The minimum accuracy observed is 35.50% for edge based experiment for Equation 4 with 2 sentence summarization for k = 2. The average of 3 sentence summarization accuracy over k, observed for the five different distances with edge based calculations are 49.00 ± 1.00%, 49.50 ± 1.41%, 49.00 ± 0.87%, 49.83 ± 0.85% and 49.83 ± 2.01%. From the result it is observed distance metrics 4 and 5 provide the same average accuracy on the test dataset.

The overall average accuracy of the five types of distance metrics on 3 sentence summarized texts and calculated using edge based formulae averaged over 'k' is 49.43 ± 0.38%, which denotes that the five distances are more or less comparable based on the observed recognition accuracies.

To analyze the result further, precision, recall and F1 measures were calculated for the bag-of-words and the best graph distance (E5):

TABLE 1.
K-NN CLASSIFICATION ACCURACIES FOR TWO-SENTENCE SUMMARIZATION

Distance metric		Value of K		
		3	5	7
$d_{mcs(\,)}$	V1	**37.50%**	37.50%	37.50%
	E1	45.00%	45.50%	**49.50%**
$d_{wgu(\,)}$	V2	40.00%	40.50%	**41.50%**
	E2	45.00%	**51.00%**	50.50%
$d_{ugu(\,)}$ / $(\lvert G_1\rvert + \lvert G_2\rvert)$	V3	37.00%	38.50%	**40.00%**
	E3	44.00%	48.00%	**50.00%**
$d_{MMCS(\,)}$ / $\lvert MCS(G_1,G_2) + mcs(G_1,G_2)\rvert$	V4	35.50%	**39.50%**	39.50%
	E4	42.50%	46.50%	**49.50%**
$d_{MMCSN(\,)}$	V5	39.50%	**42.00%**	42.00%
	E5	45.00%	49.00%	**49.50%**
bow		**50.50%**	50.00%	48.50%

TABLE 2.
K-NN CLASSIFICATION ACCURACIES FOR THREE-SENTENCE SUMMARIZATION

Distance metric		Value of K		
		3	5	7
$d_{mcs(\,)}$	V1	45.50%	45.50%	**48.00%**
	E1	47.50%	49.50%	**50.00%**
$d_{wgu(\,)}$	V2	48.00%	**48.50%**	48.50%
	E2	47.50%	**50.50%**	50.50%
$d_{ugu(\,)}$ / $(\lvert G_1\rvert + \lvert G_2\rvert)$	V3	45.00%	**46.00%**	45.50%
	E3	48.00%	**49.50%**	49.50%
$d_{MMCS(\,)}$ / $\lvert MCS(G_1,G_2) + mcs(G_1,G_2)\rvert$	V4	**46.00%**	46.00%	45.00%
	E4	49.00%	**51.00%**	49.50%
$d_{MMCSN(\,)}$	V5	47.50%	48.00%	**49.00%**
	E5	47.00%	51.00%	**51.50%**
bow		**49.50%**	49.50%	49.00%

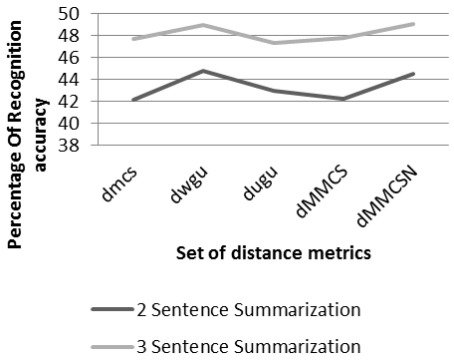


Fig 4. Recognition accuracies for vertex vs. edge based techniques

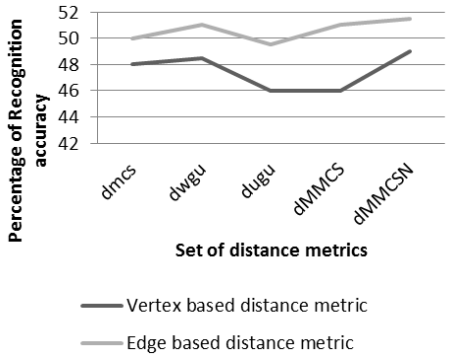


Fig 5. Recognition accuracies for 2 sentence vs. 3 sentence summarization

$$Recall = \frac{TP}{TP + FN},$$
$$Precision = \frac{TP}{TP + FP},$$
$$F\text{-}Measure = 2\frac{Recall \times Precision}{Recall + Precision}.$$

The comparative assessment of the two approaches is shown in Table 3. There it can be observed that sometimes the graph distance provides significantly better results than bag-of-words approach.

In the case of the *carcass* class the bag-of-word approach provides very satisfactory result due to having simple words like "beef" or "pork" which are enough to uniquely identify the category. On the other hand, the gold class shares some common words with other classes. The word "gold" itself can be found in copper and alum classes.

TABLE 3.
RECALL, PRECISION AND F-MEASURE FOR BAG-OF-WORDS AND GRAPH BASED SEMANTIC APPROACHES

Class	***Recall***		***Precision***		***F-Measure***	
	BOW	**Graph**	**Bow**	**Graph**	**Bow**	**Graph**
acq	0.10	0.60	0.20	0.60	0.13	0.60
alum	0.50	0.30	1.00	0.38	0.67	0.33
barley	0.60	0.50	0.46	0.50	0.52	0.50
bop	0.50	0.80	0.46	0.67	0.48	0.73
carcass	0.90	0.30	1.00	0.33	0.95	0.32
cocoa	0.60	0.60	0.86	0.86	0.71	0.71
coffee	0.50	0.50	0.46	0.46	0.48	0.48
copper	0.30	0.50	0.60	0.31	0.40	0.39
corn	0.40	0.20	0.80	0.40	0.53	0.27
cotton	0.80	0.40	0.47	0.27	0.59	0.32
cpi	0.80	0.60	0.38	0.60	0.52	0.60
crude	0.70	0.60	0.54	0.43	0.61	0.50
dlr	0.30	0.80	0.75	0.62	0.43	0.70
earn	0.40	0.80	1.00	0.89	0.57	0.84
fuel	0.30	0.50	0.19	0.39	0.23	0.44
gas	0.30	0.30	1.00	0.33	0.46	0.32
gold	0.20	0.60	0.33	0.86	0.25	0.71
grain	0.60	0.30	0.40	0.43	0.48	0.35
interest	0.50	0.30	0.28	0.60	0.36	0.40
ipi	0.90	0.80	0.75	0.80	0.82	0.80
Average	**0.51**	**0.52**	**0.60**	**0.54**	**0.51**	**0.51**

Other common words like "reserves" occur in many other classes. Moreover, there are words like "ounces" or "carat" which are overlooked in the bag-of-words approach due to their comparatively low no. of occurrences.

The use of the semantic approach enables the binding of words like "gold", "reserves", "carat" and "ounce" in such a way that they are highly unique for the gold class, giving better results. Hence, it can be strongly established that the graph distance based approach provides a much better recognition rate for textual data with semantically coherent information.

V. CONCLUSIONS AND FUTURE WORK

In the present work, we have proposed a comparative study of different graph metrics for text classification using sematic information. Our approach combines deep linguistic analysis and graph based classification techniques.

The former part of our work includes extracting discourse information from documents followed by a comprehensive similarity analysis using existent graph based distance metrics. During the calculation of the distance metrics, we have proposed an approximate version for the traditionally NP-Complete problem of finding the maximum common subgraph that is not only computationally faster but also more suited to textual similarity extraction.

Finally, we combined the graph-drs structures and the proposed distance metrics for the text classification task using a k-NN classifier. The obtained results clearly depict that the performance of most of the graph similarity metrics using our approach are more likely same. The obtained results also signify that the proposed approach is nearly equivalent to the standard bag-of-words approach. Even in some cases, it was able to outperform the approach. This result is also a good indicator of the adequacy of using semantic information to represent texts and text content.

Our future work will emphasize to analyze the impact of the summarization module in text classification task. In addition to that different machine learning algorithms, such as multi-layer perceptron, support vector machines using a graph kernel can also be applied to our proposed methodology for obtaining better results.

ACKNOWLEDGMENTS

This work was funded by Emma in the framework of the EU Erasmus Mundus Action 2.

REFERENCES

[1] S. Bleik, "Text Categorization of Biomedical Data Sets Using Graph Kernels and a Controlled Vocabulary," *EEE/ACM Trans. Comput. Biol. BioinformaticsI*, vol. 99, p. 1, Mar. 2013.

[2] L. Zhang, Y. Li, C. Sun, and W. Nadee, "Rough Set Based Approach to Text Classification," *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, vol. 3, 2013, pp. 245–252.

[3] Z. Wang and Z. Liu, "Graph-based KNN text classification," *Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on*, vol. 5, 2010, pp. 2363–2366.

[4] R. Angelova and G. Weikum, "Graph-based Text Classification: Learn from Your Neighbors," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006, pp. 485–492.

[5] H. Kamp and U. Reyle, *From Discourse to Logic: An Introduction to Model Theo-retic Semantics of Natural Language, Formal Logic and Discourse Rep-resentation Theory*. Kluwer, Dordrecht: D. Reidel, 1993, p. 717.

[6] "Graph Matching," in *Graph Classification and Clustering Based on Vector Space Embedding*, vol. Volume 77, WORLD SCIENTIFIC, 2010, pp. 15–34.

[7] M. Himsolt and G. Iversität Passau, 94030 Passau, "GML: A portable Graph File Format," 1996.

[8] H. Bunke, P. Foggia, C. Guidobaldi, C. Sansone, and M. Vento, "A Comparison of Algorithms for Maximum Common Subgraph on Randomly Connected Graphs," in *Structural, Syntactic, and Statistical Pattern Recognition SE – 12*, vol. 2396, T. Caelli, A. Amin, R. W. Duin, D. Ridder, and M. Kamel, Eds. Springer Berlin Heidelberg, 2002, pp. 123–132.

[9] J. Curran, S. Clark, and J. Bos, "Linguistically Motivated Large-Scale NLP with C&C and Boxer," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, 2007, pp. 33–36.

[10] J. Bos, "Wide-Coverage Semantic Analysis with Boxer," in *Semantics in Text Processing. STEP 2008 Conference Proceedings*, 2008, pp. 277–286.

[11] E. L. Steven Bird, Ewan Klein, *Natural Language Processing with Python*. O'Reilly Media, 2009, p. 504.

Sistema de medición de distancia mediante imágenes para determinar la posición de una esfera utilizando el sensor Kinect XBOX

Omar Rodríguez Zalapa, Antonio Hernández Zavala,
Jorge Adalberto Huerta Ruelas

***Resumen*—En este documento se presenta un método para medir la distancia del centroide de un objeto segmentado en una imagen de color con respecto a un punto de referencia fijo. El algoritmo se probó mediante una secuencia de imágenes de color, analizando más de 100 posiciones verticales diferentes de una esfera alojada en el interior de una columna cilíndrica transparente de acrílico con diámetro y longitud constante. El algoritmo propuesto integra técnicas de corrección por balance de blancos y de calibración de la cámara con sus parámetros intrínsecos, además, se prueba un nuevo método de segmentación en color utilizado para calcular distancias del mundo real a partir de imágenes en color RGB. Los resultados obtenidos reflejan una alta confiabilidad ya que el 100% de las mediciones realizadas tuvo un error menor a 1.64% con un nivel de precisión más alto que el instrumento utilizado de referencia, en un rango de distancia de 0 a 1340 mm.**

***Palabras clave*—Fotogrametría, medición de distancias en imágenes, sensor visual de distancias, metrología visual de simple vista, segmentación, seguimiento de objetos.**

Distance Measurement System using Images to Determine the Position of a Sphere using the XBOX Kinect Sensor

***Abstract*—This paper presents a method to measure the distance from the centroid of a segmented object in a color image with respect to a fixed reference point into the image. The algorithm was tested using a color image sequence by analyzing over 100 different vertical positions of a ball housed inside a transparent acrylic cylindrical column of constant diameter and length. The proposed algorithm integrates technics of correction by white balance and calibration of the camera with its intrinsic parameters; in addition, a new color segmentation method is tested to calculate real-world distances into color images RGB. The results show high reliability because 100% of measurements had a relative error in percentage less than 1.64%, with a higher level of precision than the reference instrument used in a distance range from 0 to 1340 mm.**

***Index Terms*—Photogrammetry, measuring distances into images, visual sensor of distances, simple view metrology, segmentation, tracking objects.**

Manuscrito recibido el 11 de marzo de 2014; aceptado para la publicación el 30 de mayo del 2014.

Los autores están en el Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada C.I.C.A.T.A., Unidad Querétaro, México (correos: omar.rodriguez.2013@ieee.org, anhernandezz@ipn.mx, jhuertar@ipn.mx)

I. Introducción

El desarrollo de cámaras digitales con alta tecnología y día a día con mejores capacidades, ha permitido que en diferentes aplicaciones se utilicen como una alternativa a la visión humana, en diferentes tareas como, en la supervisión de personas mediante sistemas "inteligentes" que pueden detectar, contar, identificar y seguir la trayectoria de las personas [1], [2]; en sistemas de inspección visual automatizados de productos para el control de su calidad en las empresas de manufactura [3], entre muchas otras aplicaciones que crecen día a día.

En particular, el empleo de cámaras como sensores de medición de distancia ha tenido diferentes aplicaciones. Por ejemplo, su utilización para permitir la navegación autónoma de robots terrestres [4]; en arquitectura como un instrumento de medición en interiores para obtener las dimensiones de paredes y pisos, así como para la ubicación correcta de muebles con el propósito de diseño de interiores, en exteriores para medir el tamaño y la posición de ventanas y puertas [5], y muchas más. Una imagen o secuencia de imágenes trae consigo una cantidad muy grande de información geométrica acerca de la escena representada, se han desarrollado diferentes técnicas para la construcción de escenarios 3D a partir de imágenes en 2D [6].

Se han desarrollado técnicas y métodos de descomposición de imágenes para su representación en el espacio del mundo real. Thomas Bucher [7], describe un método para mapear una imagen a coordenadas del mundo real y obtener así, una aproximación de la altura de objetos, longitudes y cambios de posición; basándose en un pequeño grupo de parámetros de fácil estimación a partir de características de los objetos o marcas en la escena, esto sin la necesidad de requerir alguno de los parámetros intrínsecos de la cámara.

Lázaro et al. [8], presentan la caracterización de la variación de intensidad de niveles de grises y su análisis mediante FFT (*Fast Fourier Transform*: transformada rápida de Fourier), en imágenes tomadas para medir la distancia entre un diodo emisor de infrarrojo y el centro de una cámara. El método propuesto se aplicó para hacer una estimación de distancias en el rango de 420 a 800 cm, logrando una exactitud sobre el 3%.

Al área de investigación que corresponde el obtener mediciones de distancias a partir de imágenes se le llama Metrología Visual y puede ser clasificada en dos tipos: *Metrología de simple vista* y *metrología de múltiples vistas* [9].

Entre las ventajas de la metrología visual sobre otras técnicas es que sólo se requiere una vista del objeto (capturada en una imagen) para hacer una medición, por lo cual se considera un método no invasivo, fácil de utilizar, con mayor cantidad de información, posibilidad de determinar muchas distancias en el sistema en base a una secuencia de imágenes y registro histórico para análisis posterior. Aunque, en ciertas aplicaciones, el resultado de la medición se requiere lo más pronto posible con respecto al momento en el que ésta se realizó, y el procesamiento digital de la imagen siempre consumirá un tiempo que se debe tomar en cuenta en el sistema de medición.

El problema de medir las dimensiones de objetos de manera directa con instrumentos como por ejemplo, cinta métrica, flexometro, regla, calibrador (Vernier), micrómetro, etc., es que el objeto tiene que estar disponible físicamente para colocar el instrumento de medición sobre él.

Existen otros métodos para medir distancias sin contacto llamados activos, que requieren la activación de un emisor para generar ya sea un ultrasonido, un rayo de luz infrarroja o un láser; éste emisor, se debe direccionar hacia un punto específico para medir mediante un receptor, las características de retorno de la señal emitida y así, poder determinar la distancia existente entre el instrumento y un punto sobre un objeto remoto. En la actualidad, éste tipo de instrumentos de medición de distancias digitales pueden traer incorporadas las funciones de registro histórico de las mediciones en chips de memoria interna o externa al instrumento, así como algunos tienen la posibilidad de comunicación con una computadora mediante puertos de comunicación como RS232, USB o Ethernet.

En metrología visual de múltiples vistas, se requieren dos o más cámaras dispuestas espacialmente en una forma particular, y se requiere conocer las propiedades específicas de cada cámara individual para obtener mediciones exactas. El hecho de que se necesite tomar y analizar más de dos imágenes de la misma escena también hace que el sistema sea más complejo de programar y de usar. En este trabajo se determinó utilizar *metrología visual de simple vista* ya que implica mayor facilidad y menor costo económico y computacional.

En la sección II, se presenta la descripción del problema y la arquitectura de los elementos que intervienen para medir la distancia que hay del centroide de una esfera dentro de una columna cilíndrica de acrílico a la base de la misma, mediante la utilización de imágenes de color. En la sección III, se explica la propuesta de solución. En la sección IV, se muestran los resultados obtenidos y finalmente en la sección V, se presenta algunas conclusiones.

II. Descripción del problema

Se requiere diseñar e implementar un sistema de medición para determinar la distancia que hay entre una esfera, que se ubicará en posiciones fijas y estables, a lo largo de una columna de 1395 mm de longitud, y un punto de referencia fijo, llamado base de la columna. El desplazamiento incremental vertical controlado de la esfera se llevó a cabo mediante un hilo amarrado a ella que salía por la parte superior de la columna y la mantenía suspendida en la posición deseada. Una vez validada la precisión y exactitud del sistema de medición, se integrará a un sistema automático de control de posición de una esfera levitada neumáticamente.

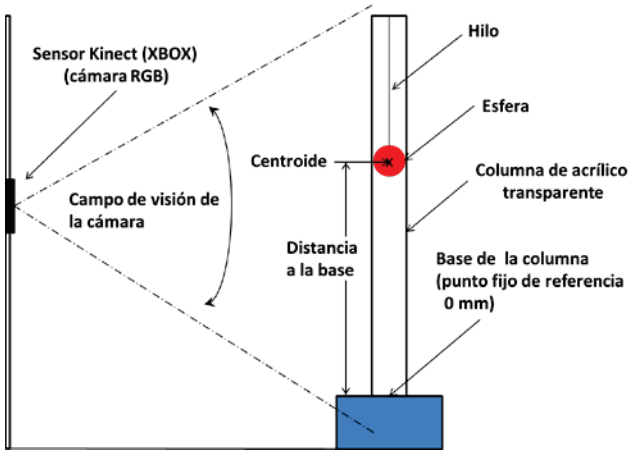


Fig. 1. Arquitectura del sistema de medición.

En la fig. 1, se muestra la localización de cada uno de los componentes del sistema visual de medición de distancias. La captura de imágenes de color se llevó a cabo mediante el sensor Kinect XBOX 360 por dos razones principales: a) Bajo costo (en comparación con una cámara digital industrial con similares características); b) Configuración ágil y simple para que pueda ser utilizado con una computadora mediante un cable USB (se requiere una fuente externa de alimentación para la cámara). Además, se cuenta con un soporte robusto mediante el acceso abierto a diferentes tipos de drivers, librerías y desarrollo de aplicaciones listas para ser utilizadas y con acceso libre al código fuente, todo esto para diferentes plataformas como Mac, UNIX, y Windows.

Las características destacadas del sensor XBOX 360 de Microsoft para éste proyecto se muestran en la tabla I. Las

características completas de éste dispositivo se han publicado en diferentes documentos [12, 13].

En la tabla I, se indica que el campo de visión del sensor es de 57° en horizontal y 43° en vertical, considerando que el sensor se dispone en una posición horizontal como la que se muestra en la fig. 2. Para el sistema de medición se determinó colocar el sensor en dirección vertical como se muestra en la fig. 1, obteniendo de esta manera un campo de visión sobre el eje vertical de 57°. Se ajustó la posición definitiva del sensor sobre un tubo vertical de metal de tal manera, que su campo de visión cubriera la totalidad de la columna de acrílico por la que se desplaza la esfera.

TABLA I.
CARACTERÍSTICAS PRINCIPALES DEL SENSOR KINECT XBOX 360

Característica	Valor
Campo de visión angular	57 ° horizontal, 43° vertical
Rango de inclinación física	± 27 °
Máximo valor de flujo de datos	Aprox. 30 cuadros por segundo
Resolución de imágenes de color	640 × 480 pixeles (VGA)
Tipo de conexión del dispositivo	USB (+ fuente externa de energía)

La columna cilíndrica de acrílico transparente se encuentra separada a una distancia de 1425 mm del centro de la cámara, y las dimensiones de la columna cilíndrica son 1395 mm de altura con un diámetro externo de 76.2 mm y un grosor de paredes de 3 mm; de manera externa la mitad de la columna cilíndrica se cubrió con una película de vinil negro, dejando el lado descubierto en dirección del eje focal de la cámara.

Fig. 2. Posición horizontal del sensor Kinect XBOX 360® [14].

La base tiene dimensiones de 299x248x295 mm y es del mismo material que la columna. La esfera tiene un diámetro de 60 mm y es de poliestireno expandido (unicel) pintada de color rojo, está alojada dentro de la columna de acrílico y puede deslizarse libremente en dirección vertical. En la parte inferior interna de la base de la columna se alojara como trabajo posterior una tarjeta de control de velocidad para un motor de c.d. (corriente directa) con aspas, que inyectara aire para elevar la esfera dentro de la columna. El aíre saldrá por la parte superior, y a su vez se colocara una trampa para que no se salga la esfera.

Para probar y validar la exactitud y precisión del sistema de medición de distancias mediante imágenes de color, se consideraron condiciones estables y controladas para la ubicación de la esfera en posiciones fijas especificas a lo largo de la longitud de la columna, esto se logró mediante la suspensión de la esfera con un hilo amarrado a ella, cuya longitud se ajustaba y éste salía por la parte superior de la misma; como se puede observar en la fig. 1.

III. PROPUESTA DE SOLUCIÓN

De manera previa a la captura de imágenes y su procesamiento, se ajustó la orientación de la columna y del sensor a un eje vertical de 90° y la base se niveló con respecto a un eje horizontal de 0°.

Para poder realizar una medición de distancia en una secuencia de imágenes capturadas por el sensor, se debe considerar el modelo de cámara oscura (*pinhole*) [15] del plano de la cámara que se encuentra representado en la fig. 3. Aquí se muestra que un punto **X** en el espacio real 3D, es representado en el plano de la imagen como **x**. Las coordenadas Euclidianas *(X, Y, Z)* definen la ubicación del punto **X** en el espacio real, así como *(x, y)* definen la posición del mismo punto en el plano de la imagen en la cámara denotado por **x**.

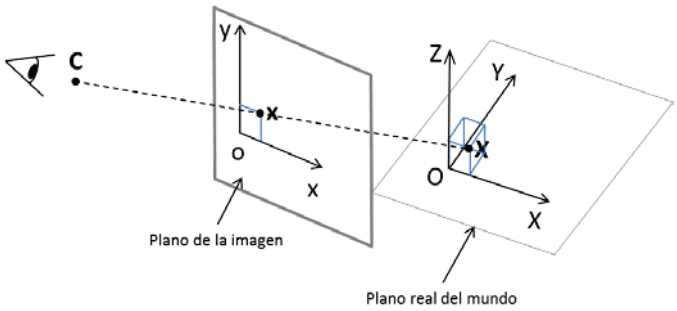


Fig. 3. Modelo "*pinhole*" (o cámara oscura) que representa el plano 2D de la imagen en la cámara con respecto al espacio real 3D del mundo. Un punto X en el espacio real es generado como x en el plano de la cámara. El punto C representa el centro de la cámara [15].

El proceso de calibración para obtener los parámetros intrínsecos y extrínsecos de la cámara de color incorporada en el sensor, se llevó a cabo siguiendo la metodología del ToolBox para MatLab desarrollado por Jean-Yves Bouguet [16]; obteniendo los siguientes resultados con incertidumbres de los parámetros intrínsecos (modelo de la cámara).

Longitud Focal:

fc = [523.95439 520.91487] ± [2.45522 2.02515]

Punto principal:

cc = [319.50000 239.50000] ± [0.00000 0.00000]

Sesgo:

$alpha\_c$ = [0.00000] ± [0.00000] ⇒

ángulo de pixel ejes = 90.00000 ± 0.00000 *grados*

Distorsión:

kc = [0.13660 -0.29048 0.00217 -0.00161 0.00000] ± [0.02046 0.06266 0.00137 0.00108 0.00000]

Error de pixel: err = [0.83738 0.83172]

Los valores mostrados de los parámetros intrínsecos de la cámara son los que directamente arroja el software de calibración con formato propio del autor. Y estos valores son los que se tomaron en cuenta para llevar a cabo las correcciones necesarias en las imágenes capturadas. Además, debemos considerar otras operaciones de ajuste y calibración inicial tomando en cuenta alguna imagen *muestra* del escenario real de medición, como la que se indica en la fig. 5 (a), que muestra el objeto de estudio inmerso en el escenario real con cierta inclinación inducida que el programa de computadora desarrollado corregirá. En la fig. 4. se muestra el diagrama de flujo del algoritmo completo del proceso de ajuste inicial para realizar la medición de distancias en imágenes de color.

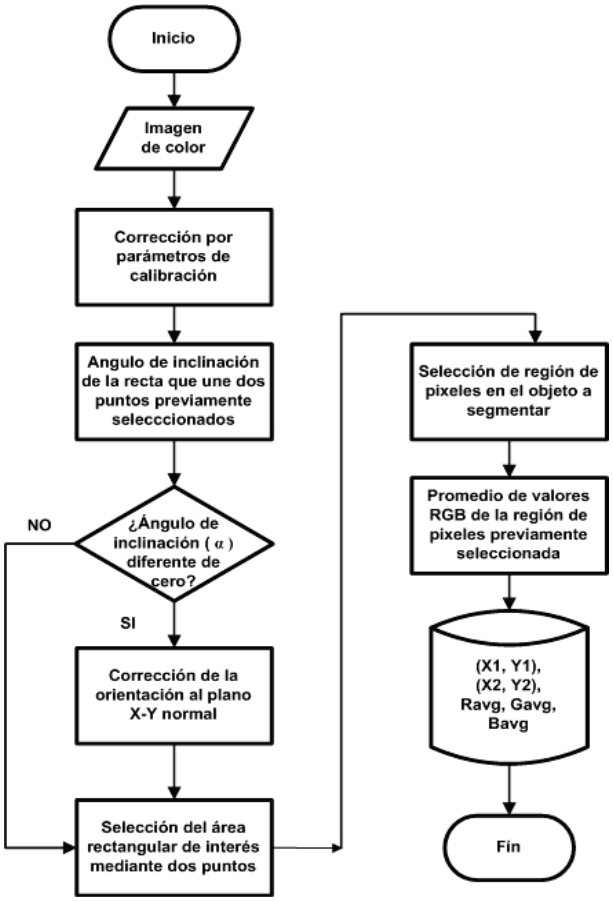
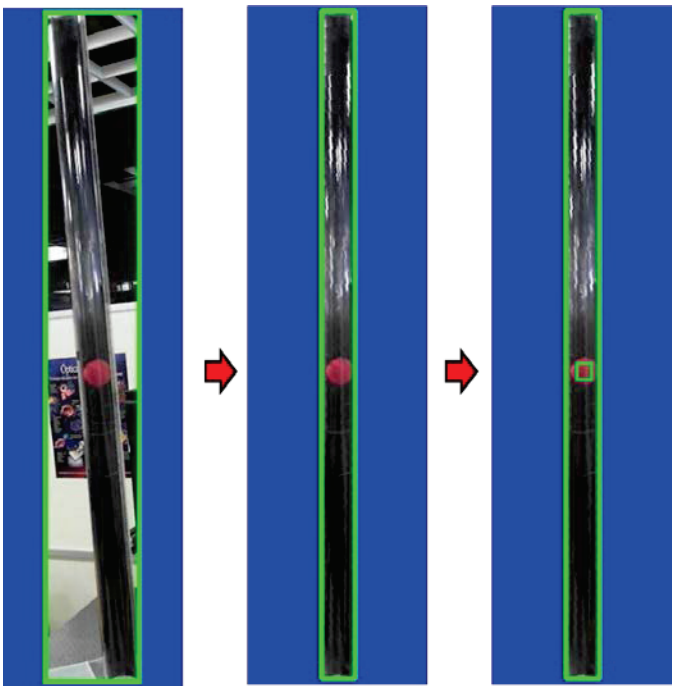


Fig. **4**. Diagrama de flujo del proceso de ajuste inicial para realizar medición de distancias en imágenes de color.

Se consideró incluir como parte del ajuste inicial una operación para determinar la posible desviación del plano horizontal con respecto a 0°, que pudieran tener objetos característicos de una imagen muestra del escenario real de medición. Para llevar a cabo esto, el usuario selecciona dos puntos sobre un borde de línea recta de algún objeto que se supone está en el plano horizontal (0°), y se calcula el ángulo de inclinación (α) de la línea recta que une a estos puntos como se indica en el diagrama de flujo mostrado en la fig. 4. Si éste ángulo es diferente de cero se realiza una corrección de la imagen mediante una rotación plana, en magnitud y sentido indicado por el ángulo α.



(a) (b) (c)

Fig. 5. Se muestran las operaciones de ajuste inicial: (a) y (b) Corrección de la orientación al plano X-Y normal y selección del área rectangular de interés y c) selección de región de pixeles en el objeto a segmentar.

Posteriormente el usuario selecciona manualmente el área rectangular de interés (área que encierra la trayectoria de desplazamiento del objeto: ver la fig. 5 (b)). De la misma manera, se selecciona una región rectangular de pixeles en el objeto a segmentar, para obtener el promedio de los valores RGB (ver la fig. 5 (c)). El resultado del proceso completo del ajuste inicial se muestra en la fig. 5, por cuestión de presentación, las tres imágenes con resolución de 640x480 pixeles se han girado 90° y recortado en el mismo factor de ancho y alto quedando con un valor de 4.5 cm de ancho por 17 cm de alto.

La técnica de segmentación propuesta extraerá el objeto de interés en una secuencia de operaciones sobre cada imagen de entrada. Considerando que hay variación continua en la uniformidad de la intensidad luminosa (debido a variación de la luz natural del medio ambiente, variación de la luz artificial de lámparas, apertura y cierre de puerta, así como, diferentes niveles de luz reflejada en superficies lisas reflejantes como vidrios, piso, mesas, etc. y en la propia columna cilíndrica de

acrílico) que afecta el color de las diferentes regiones sobre la imagen, provocando intersecciones entre regiones adyacentes; el algoritmo de segmentación desarrollado resolverá esto, para extraer con una buena fidelidad el objeto de interés.

Una vez realizado el proceso de ajuste inicial, el proceso para determinar la distancia del centro de la esfera a la base de la columna, es el que se indica en el diagrama de flujo mostrado en la fig. 6. En él se indica que una vez que una imagen es capturada (posterior al ajuste inicial), se realizan las siguientes tres operaciones de corrección:

a) Corrección por parámetros de calibración intrínsecos de la cámara de color.
b) Corrección por balance de blancos. Consiste en realizar un balance automático de tonos blancos sobre la imagen, dado que la intensidad luminosa y el tono de diferentes fuentes de luz afecta el color de los objetos [17] y que el cerebro humano y la retina son capaces de percibir y determinar el color de un objeto bajo diferentes condiciones de iluminación, llamándole a esta habilidad *constancia de color* [18, 19].
c) Corrección de la orientación de la imagen al plano X-Y normal. Se aplica sólo si en las operaciones de ajuste inicial, se determinó un ángulo α de desviación diferente de 0°.

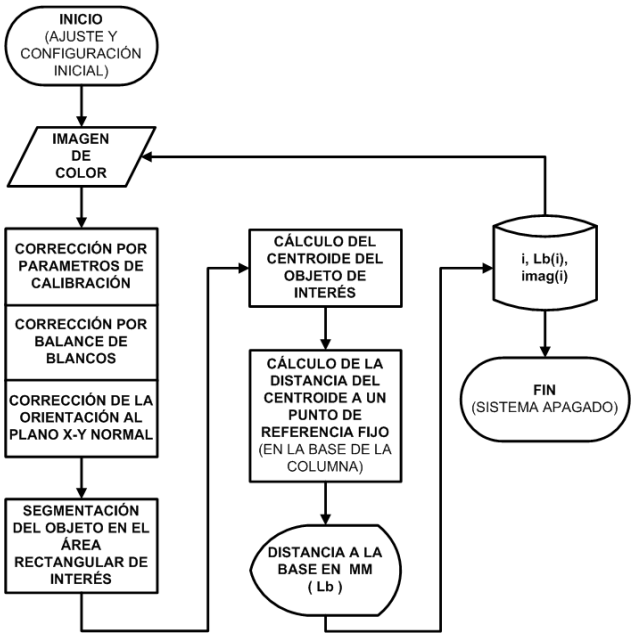


Fig. 6. Diagrama de flujo del proceso de medición de distancia en imágenes de color.

El sensor XBOX Kinect no trae incorporado el autobalance de blancos para las imágenes RGB capturadas mediante su cámara de color, por lo que se elaboró un programa para que en cada una de imágenes capturadas se realice el balance automático de blancos y así obtener los colores más aproximados a colores que correspondan al mundo real, mismo que es llamado desde el programa principal de procesamiento de imágenes, que lleva a cabo el cálculo de la distancia del centroide de la esfera a la base de la columna. Los sub-programas (funciones) y el programa principal fueron desarrollados en MatLab versión 7.10.0.499 (R2010a).

La selección y extracción automática del área de interés en la imagen de color, para llevar a cabo el proceso de segmentación de la esfera, se realiza tomando en cuenta las coordenadas de los dos puntos seleccionados en el ajuste inicial. El algoritmo de segmentación se lleva a cabo en los siguientes pasos (que forman parte del algoritmo mostrado en la fig. 6):

1. Calcular en el área de interés, la distancia euclidiana entre los valores *RGB* de cada pixel '*z'* en la posición (*x*, y), y los valores promedio *RGB* de los pixeles '*a'* obtenidos en el ajuste inicial.

$$D\left(Z_{RGB},\ a_{RGBavg}\right) = \tag{1}$$
$$= \sqrt{\left[\,(z_R - Ravg)^2 + (z_G - Gavg)^2 + (z_B - Bavg)^2\,\right]},$$

donde Z_{RGB} son los valores *RGB* de cada pixel '*z'* en la posición (*x*, y), a_{RGBavg} son los valores promedio *RGB* de los pixeles 'a' obtenidos en el ajuste inicial. Así también z_R, z_G, y z_R son los valores correspondientes de los canales Rojo, Verde y Azul del pixel 'z' en la posición (*x*, *y*) en la región de interés y $Ravg$, $Gavg$, $Bavg$ son los valores promedio de los canales Rojo, Verde y Azul de la subregión del objeto a segmentar seleccionada en el ajuste inicial.
2. Si el valor de intensidad rojo (R) es mayor que los correspondientes valores verde (G) y azul (B) en cada posición *x, y* del pixel en el área de interés, y se cumple además que la distancia de éstos valores *RGB* a los valores promedio RGB_{avg} (calculada en el paso anterior) sea menor que un valor de umbral preestablecido; se verifica entonces que además la distancia del canal rojo al canal verde y azul en cada pixel en dicha posición sea mayor a un valor de umbral preestablecido. Si se cumplen estas tres condiciones, los valores *RGB* del pixel en la posición (*x*, *y)* se deja sin cambio y en caso contrario se les asigna una valor de cero; el resultado se convierte a una imagen binaria, esto se ilustra en la fig. 7 (a).
3. Se aplican funciones predefinas de MatLab para extraer las características de detección de elementos conectados y obtener así, el elemento conectado con un área mayor a un valor preestablecido para posteriormente, aplicar técnicas morfológicas de erosión y dilatación para eliminar artefactos de ruido en los objetos segmentados. El resultado logrado en este paso se ilustra en la fig. 7 (b) y 7 (c) respectivamente.
4. Para el objeto segmentado en los pasos 2–3 descritos anteriormente, se determina la posición (x, y) del

centroide de la esfera. En la fig. 7 (c), se puede observar una cruz sobre el objeto segmentado (esfera) dentro de la columna, indicando con una cruz blanca la posición del centroide.

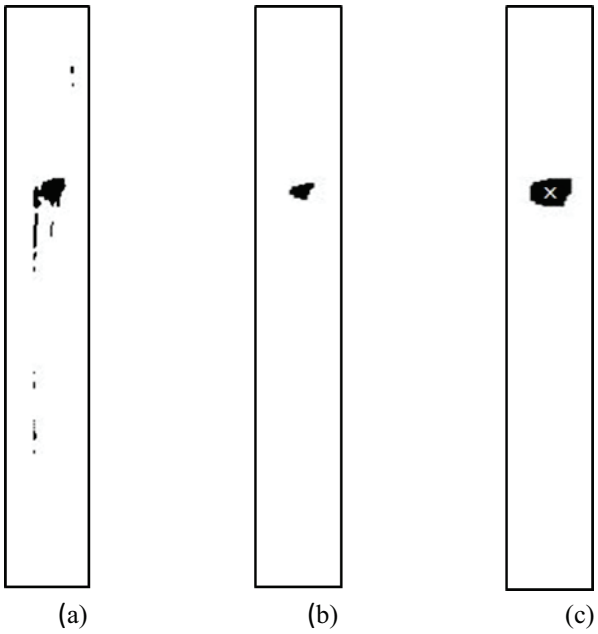


Fig. 7. Resultado de la secuencia del algoritmo de segmentación y localización del centroide: a) 1ª. Fase, discriminación de pixeles y binarización de imagen, b) 2ª. Fase, Selección de mayor área de elementos conectados y erosión, c) 3ª. Fase, Dilatación y localización del centroide

Finalmente se determina la distancia en milímetros que hay del centroide del objeto segmentado a un punto de referencia fijo en la base de la columna. Esto se lleva a cabo, estableciendo una relación de proporcionalidad entre la distancia medida en pixeles del centroide al extremo de la base de la columna y la longitud total de la columna en pixeles y en mm; conociendo estos valores a partir de las operaciones de ajuste inicial del sistema de medición. Se imprimen y se guardan como resultados la distancia del centroide de la esfera a la base de la columna, el número de imagen y su nombre.

Como se indica en la fig. 6, el ciclo se mantiene en forma continua y empieza de la misma manera cuando se ha capturado una nueva imagen. El proceso de captura y análisis se detiene cuando ya no hay imágenes capturadas o el sistema se apaga.

IV. Resultados obtenidos

Se capturaron 134 posiciones fijas incrementales de la esfera con intervalos de 10 mm a partir de la base de la columna (posición 1: 0 mm) y hasta una altura de 1340 mm (posición 134). Cada posición representó un experimento individual de medición en el que se tomaron 5 imágenes de color RGB, para su procesamiento mediante el algoritmo propuesto y poder determinar así, la distancia del centroide de la esfera a la base de la columna.

Para ubicar la esfera en la posición fija deseada y como referencia de comparación de las mediciones realizadas, se utilizó un Telémetro Láser Bosch DLE40 [20], que se nivelo y ajusto para que quedara de manera fija y estable en la parte inferior de la columna dentro de su base, manteniendo las mismas condiciones de operación del instrumento entre cada lectura de distancia. Una vez que la esfera se posicionaba en forma manual, en la distancia deseada (ajustando la longitud del hilo que permitió suspender la esfera y mantenerla de manera estable en la altura deseada y con ayuda de la lectura del telémetro), el sensor Kinect recibía la orden de capturar 5 imágenes y se procesó cada una de manera independiente para obtener un valor de distancia (mediante el algoritmo mostrado en la fig. 6); así también, por cada bloque de cinco imágenes se registraron también las cinco mediciones correspondientes con el telémetro láser para cada una de las posiciones de la esfera, este proceso se repitió de manera secuencial hasta completar 134 × 5 mediciones, correspondientes al número total de posiciones de la esfera.

Fig. 8. La vista frontal del Telémetro Láser Bosch DLE40 [20].

Al finalizar el proceso, se almacenó en un archivo de texto el resultado de las mediciones, en éste archivo se registró el número de la medición, los cinco valores de distancia correspondientes a cada una de las imágenes capturadas y su promedio correspondiente a cada posición de la esfera; para su análisis posterior. En la fig. 8, se muestra una foto de la vista frontal del instrumento de medición de referencia y en la tabla II, se indican sus características de medición así como algunas especificaciones técnicas. Para una referencia completa consultar [20].

Como se menciona en el manual de operación [20], el aparato de medición ha sido proyectado para medir distancias en forma manual de longitudes, alturas, separaciones, y para calcular superficies y volúmenes. El aparato de medición es adecuado para trazar medidas en la construcción tanto en interiores como en exteriores. En el proyecto se utilizó como referencia para verificar el grado de precisión y exactitud de las mediciones de distancia en imágenes RGB.

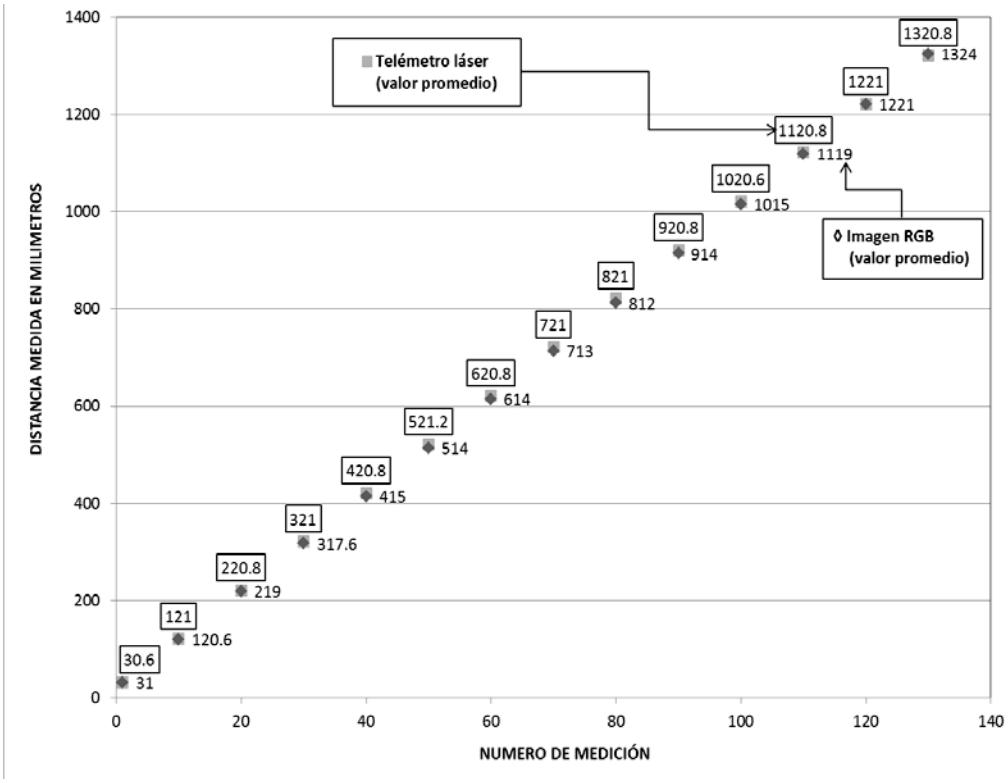


Fig. 10. Grafica comparativa de las mediciones de distancia realizadas con imágenes RGB, con respecto a las mediciones realizadas con el Telémetro Láser Bosch DLE40; mostrando 14 posiciones representativas de un total de 134.

En la fig. 9, se muestran los resultados obtenidos al realizar mediciones independientes una de otra de 134 posiciones diferentes de la esfera dentro de la columna; la gráfica muestra 14 posiciones representativas de la trayectoria vertical completa.

TABLA II.
CARACTERÍSTICAS DE MEDICIÓN Y ESPECIFICACIONES TÉCNICAS DEL TELÉMETRO LÁSER BOSCH DLE40.

Especificación	Valor
Clase del láser	2
Tipo de láser	635 nm, < 1mW
Rango de medición (en interiores)	0.05 – 40 m
Precisión de medición (típica)	± 1.5 mm
Pilas de consumo	4 x LR03 (AAA) (incluidas)
Peso aprox. con batería	0.18 kg
Dimensiones	58 x 100 x 32 mm

En el eje X se muestra el número de la medición realizada individualmente, en cada posición y en el eje Y se muestra el valor de la distancia medida de dicha posición en milímetros (mm) utilizando en su representación el valor promedio del resultado de las 5 imágenes RGB y de la misma manera el valor promedio de las 5 lecturas del Telémetro Láser respectivamente.

Como se puede observar las mediciones de distancia hechas mediante las imágenes RGB se aproximan a las lecturas tomadas en el instrumento. Como se observa en la gráfica, los valores promedio correspondiente al telémetro láser se muestran encerrados en un recuadro para diferenciarlos de los valores correspondientes al resultado de distancia promedio medida mediante las imágenes RGB.

Se calculó la magnitud del error relativo porcentual para cada una de las 134 posiciones diferentes mediante la ecuación (2),

$$\% E_{rel.} = \frac{E_{abs.}}{V_{real}} \times 100 = \frac{|V_{real} - V_{med.}|}{V_{real}} \times 100 \qquad (2)$$

donde $\% E_{rel.}$ es igual al porciento de error relativo, $E_{abs.}$ es el error absoluto que se obtienes al calcular la diferencia absoluta entre el valor real de referencia (V_{real}) y el valor medido ($V_{med.}$), en nuestro caso, el valor real de referencia es la lectura de distancia del telémetro laser y el valor medido es la distancia resultante en la imagen RGB.

En la gráfica representada en la fig. 11, se puede observar la distribución poblacional del error relativo porcentual de las 134 mediciones realizadas independientemente una de otra. Se puede notar como se tiene una desviación máxima de 1.64%. El 100% de las mediciones realizadas tuvieron un porciento de error relativo que se mantuvo en un rango del 0 a 1.64%; en un rango de distancia de 0 a 1340 mm sobre el eje Z, paralelo al plano de imagen en la cámara (ver la fig. 3).

Las magnitudes del error absoluto y del error relativo porcentual es variable de una medición a otra ya que hay factores no controlados que afectan las mediciones: a) Nivel

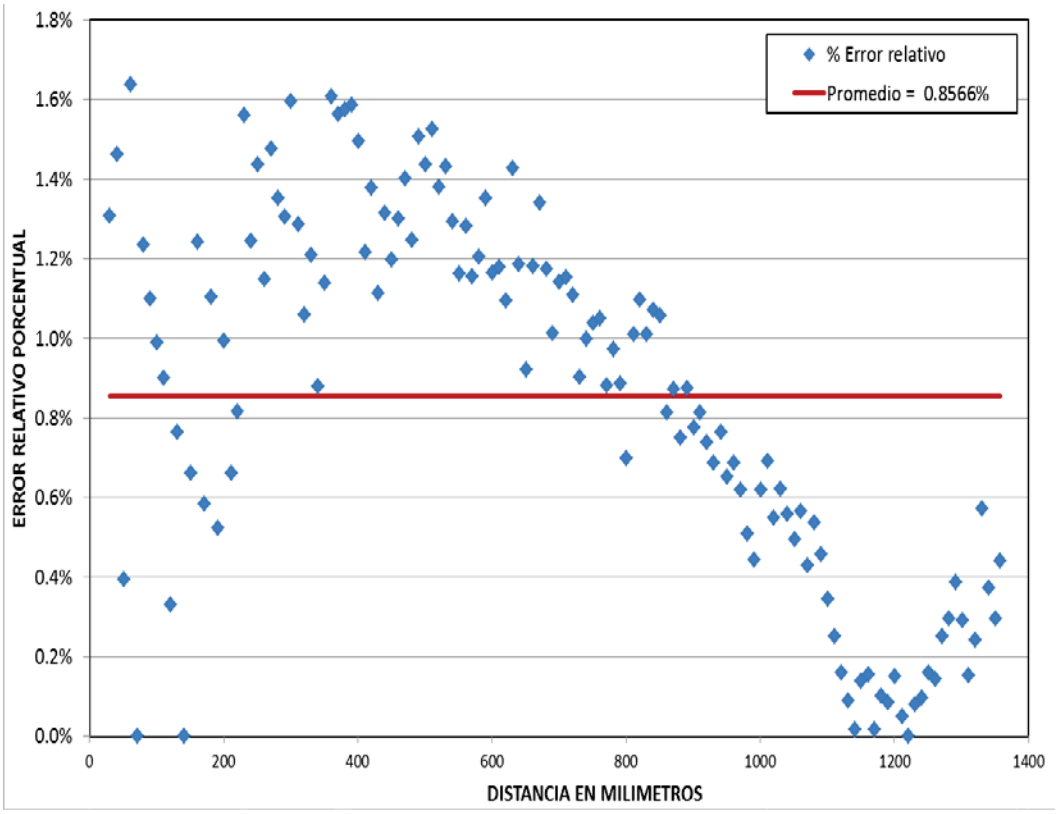


Fig. 11. Grafica que muestra la distribución poblacional del error relativo porcentual para un total de 134 mediciones realizadas.

de iluminación de escenario, b) reflexión y refracción no homogénea de la luz en la columna de acrílico transparente ya que el objeto de interés se encuentra dentro de ésta, y c) exactitud y precisión del instrumento de referencia (Telémetro Láser) al realizar cada medición.

En la tabla III, se muestran los valores máximo y mínimo del total de las desviaciones típica estándar (calculadas para cada 5 mediciones correspondientes a cada posición de la esfera), utilizando ambos métodos, el telémetro láser y el sistema de medición mediante imágenes RGB.

TABLA III.

VALOR MÁXIMO Y MÍNIMO DE LA DESVIACIÓN TÍPICA ESTÁNDAR, OBTENIDAS DE 134 POSICIONES DE LA ESFERA Y REALIZANDO 5 MEDICIONES POR POSICIÓN, UTILIZANDO AMBOS MÉTODOS: TELÉMETRO LÁSER Y SISTEMA MEDIANTE IMÁGENES RGB, INDICANDO TAMBIÉN LA OCURRENCIA DE ESTOS VALORES. LA σ REFIERE AL VALOR DE DESVIACIÓN TÍPICA ESTÁNDAR.

Método	Máximo σ	Ocurrencia (repetición del valor)	Mínimo σ	Ocurrencia (repetición del valor)
Imágenes RGB	1.96	2	0	109
Telémetro láser	0.49	21	0	74

Como se observa en la tabla III, el máximo valor de desviación típica estándar es mayor en el método de imágenes RGB que en el método de telémetro láser pero solamente se obtuvo este valor en dos grupos de mediciones correspondientes a dos posiciones; mientras que utilizando el método del telémetro la ocurrencia de la máxima desviación típica estándar fue de 21. Y el mínimo valor de desviación típica estándar en ambos métodos fue de cero, sin embargo, fue más alta en el método de imágenes RGB en un 47.3% con respecto al método del telémetro láser. Indicando con esto que, el método de imágenes RGB tiene una precisión más alta que el método de telemetro láser para medir distancias bajo las mismas condiciones del experimento realizado.

V. COCLUSIONES

El sistema de medición propuesto para determinar la posición de la esfera, dio como resultado que el 100% de las mediciones realizadas tuvieron un error relativo porcentual en un margen de 0 a 1.64%, con un buen nivel de precisión. Debido a esto, se han generado buenas expectativas para su incorporación en sistemas de monitoreo y control automático de procesos.

Para la integración de éste sistema de medición a un sistema automático de control de posición de una esfera levitada neumáticamente, se deben tomar en cuenta los siguientes factores en su diseño: a) La esfera se encontrará en constante movimiento y su posición no se mantendrá fija, (aun cuando la velocidad del motor se mantenga constante); b) la estabilidad en una posición deseada dependerá del grado de turbulencia del aire a su alrededor y c) el movimiento será

errático y oscilante (girará alrededor de su centro de gravedad en diferentes direcciones y se moverá con desplazamientos en los tres ejes *X*, *Y*, *Z*), para una velocidad definida del motor de c.d. Por lo que, como trabajo futuro es importante realizar el análisis del desempeño de éste sistema de medición de distancia en el entorno dinámico real en el que se integrará.

REFERENCIAS

[1] H. M. Dee, S. A. Velastin, "How close are we to solving the problem of automated visual surveillance?," *Machine Vision and Applications*, vol. 19, no. 5–6, 2008, pp. 329–343.

[2] P. Vera, D. Zenteno, J. Salas, "Counting Pedestrians in Bidirectional Scenarios Using Zenithal Depth Images," *Proceedings of 5th Mexican Conference, MCPR*, Querétaro, Mexico, junio 26–29, 2013, pp. 84–93.

[3] R. T. Chin, C. A. Harlow, "Automated visual inspection: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 6, 1982, pp. 557–573.

[4] E. Royer et al., "Monocular vision for mobile robot localization and autonomous navigation," *International Journal of Computer Vision,* vol. 74, no. 3, 2007, pp. 237–260.

[5] A. Criminisi, I. Reid, A. Zisserman, "A plane measuring device," *Image and Vision Computing*, vol. 17, no. 8, 1999, pp. 625–634.

[6] Y. Wan et al., "A Study in 3D-Reconstruction Using Kinect Sensor," *8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, IEEE*,* 2012. pp. 1–7.

[7] T. Bucher, "Measurement of distance and height in images based on easy attainable calibration parameters," *Proceedings of the IEEE in Intelligent Vehicles Symposium, IV*, 2000. pp. 314–319.

[8] J. L. Lázaro et al., "Sensor for distance estimation using FFT of images." *Sensors*, 2009, vol. 9, no. 12, p. 10434–10446.

[9] A. Criminisi, *Accurate visual metrology from single and multiple uncalibrated images*. Springer, 2001.

[10] P. Wright, "Single View Metrology: Measuring the Dimensions of Objects From a Single Image." University of York, Computer Science Department, 2006.

[11] R. J. Moffat, "Describing the uncertainties in experimental results," *Experimental thermal and fluid science*, vol. 1, no. 1, 1988, pp. 3–17.

[12] MSDN, Microsoft, http://msdn.microsoft.com/en-us/library/jj131033, consultado el 30/08/2013.

[13] Kinect, http://wiki.ros.org/kinect, consultado el 23/01/2012.

[14] Wikipedia, http://commons.wikimedia.org/wiki/File:Xbox-360-Kinect-Standalone.png, recuperado el 15/06/2013.

[15] A. Criminisi, I. Reid, A. Zisserman, "A plane measuring device," *Image and Vision Computing*, vol. 17, no. 8, 1999, pp. 625–634.

[16] J.-Y. Bouguet. "Camera calibration toolbox for Matlab", http://www.vision.caltech.edu/bouguetj/calib_doc/2004.

[17] R. Ramanath et al., "Color image processing pipeline," *Signal Processing Magazine*, IEEE, vol. 22, no. 1, 2005, pp. 34–43.

[18] S. Zeki, Semir. *A Vision of the Brain*. Blackwell scientific publications, 1993.

[19] M. Ebner. *Color constancy*. Wiley, 2007.

[20] Bosch DLE40. http://www.boschmexico.com.mx/telemetro-laser-dle-40-professional, recuperado el 01/04/2013.

[21] R. Doskocil et al., "Measurement of distance by single visual camera at robot sensor systems," *MECHATRONIKA, 15th International Symposium,* IEEE, 2012.

Information Extraction in Semantic, Highly-Structured, and Semi-Structured Web Sources

Víctor M. Alonso-Rorís, Juan M. Santos Gago, Roberto Pérez Rodríguez, Carlos Rivas Costa, Miguel A. Gómez Carballa, and Luis Anido Rifón

***Abstract*—The evolution of the Web from the original proposal made in 1989 can be considered one of the most revolutionary technological changes in centuries. During the past 25 years the Web has evolved from a static version to a fully dynamic and interoperable intelligent ecosystem. The amount of data produced during these few decades is enormous. New applications, developed by individual developers or small companies, can take advantage of both services and data already present on the Web. Data, produced by humans and machines, may be available in different formats and through different access interfaces. This paper analyses three different types of data available on the Web and presents mechanisms for accessing and extracting this information. The authors show several applications that leverage extracted information in two areas of research: recommendations of educational resources beyond content and interactive digital TV applications.**

***Index Terms*—Information extraction, web data processing, semantic enrichment, data mining, web scraping.**

I. Introduction

PRESENTLY, the Web hosts billons of pieces of information of all kinds and origins. A relevant part of this information can be freely accessed and reused. Under these assumptions, it would be possible to collect these freely shared pieces of information to enrich any database, that is, to complete the information stored locally using the huge amount of data available on the Web. This will enable software agents to increase their knowledge from a wide range of sources providing different points of view and levels of detail. This way, potentially complete information may be obtained.

Section II explains different strategies for information extraction depending on the type of source. Sources available may be classified according to the language used to expose this information (e.g., HTML, XML, SQL, RDF, JSON). Each of these types of sources has been designed to satisfy a different need (e.g., to be accessible to humans, to facilitate the communication between applications, to facilitate data storage). Enriching a system's database means taking the most of the available information sources online to complete information stored locally. Insofar automated enrichment is concerned, the most interesting repositories are those specifically designed to be easily readable and interpretable by machines, that is, those offering their information according to a Knowledge Representation language like the ones proposed by the Linked Open Data (LOD) initiative [1]. The reason for that is because the enrichment process' complexity is dramatically reduced, and therefore its automation is greatly simplified. On the other side, other types of sources would need a previous translation or transformation process to represent them according to a normalized language, such as RDF. This latter process will require the active participation of experts.

Section III describes the application of information extraction techniques in the context of the iTEC project, which is the flagship FP7 project in the education area, financed by the Europan Comission with 12 million euros. iTEC tries to contribute to the conception of the classroom of the future, in which technology is complemented with the most innovative pedagogical approaches, which entail a major level of dynamism in the educational practice.

Section IV briefly discusses the use of information extraction techniques in two projects that have to do with the access to services through digital TV interfaces. The first one, Berce TV, is a portal that enables parents to access information on their children in early chilhood education centers. The second one, Emprego, is a project aimed at providing an accessible job search interface though the TV.

Finally, we give conclusions and discuss future work.

II. Information Extraction from the Web

In the Internet, you can find many different sources of information, being a lot of them available for free. The nature of the information that you can freely access to is very heterogeneus. From a technical point of view—and for convenience—, we make a classification of the sources of information by the format they use for encoding information[1][2]—which many times is associated with a particular protocol for accessing to that information. The

Manuscript received on January 7, 2014; accepted for publication on February 28, 2014.

The authors are with Department of Telematics, University of Vigo, Spain (e-mails: {victor.roris, Juan.Santos, carlosrivas, miguelgomez, lanido}@det.uvigo.es, robertoperezrodriguez@gmail.com).

[1]Some of those formats are, among others, RDF, HTML, XML, and JSON.

[2]This idea is not new. See for instance the work in [2].

rationale for having so many different formats for encoding information is that each one is best suited for satisfying a particular need. For instance, RDF is well suited to be accessed from systems that belong to the so-called semantic Web; whereas HTML is particularly well suited to encode information in such a way that a Web browser is able to understand and display it in a computer's screen.

It becomes thus clear that, in order to populate a local database using external sources of information, the more semantic the source format is the less processing we need to perform at the local side. Therefore, purely semantic sources that share information in RDF format are by far the preferrable ones, whereas sources that are meant to represent markup—for presentation—are the less desirable ones, because they need much more processing in order to squeeze them and extract the "semantic juice" out of them. Most of the times, that task is very labor-intensive, and no good automated procedures are yet available.

Below, we describe the three types of sources of information: semantic sources, highly-structured sources, and semi-structured sources.

A. Semantic Sources

The kind of sources of information from which we most easily can extract information are those pertaining to the Linked Open Data (LOD) initiative. The main purpose of the LOD project is to enrich the traditional Web by publishing open datasets—made of data items—which include links to data items in different datasets. Since the purpose of that format for representing information is that it must be easily understandable by machines, those sources are therefore very convinient for extracting information.

In order to be compliant with Linked Open Data conventions, information is represented in Resource Description Framework (RDF) format. Despite the fact that it was meant as a format for representing metadata, RDF is used—in the context of LOD—for modelling general information. The simple model for representing information that RDF implements accounts for its success in representing information beyond the limits of the Semantic Web[3]. RDF stores information as triples, which format every piece of information as an statement composed of subject, predicate, and object.

Information stored in RDF triples and published in public access points is commonly accessed by using the SPARQL query language—in fact, those public access points are frequently called SPARQL endpoints [3]. With SPARQL we can compose queries that are able to retrieve and manipulate data stored in those sources that comply with LOD conventions.

In addition to those sources, there exist a number of services that allow for launching searches on information stored in the LOD network, much like Google does with the traditional Web. Those tools, such as Sindice [4] and SWSE [5] are very useful and enable us to discover a lot of interesting sources of information.

B. Highly-structured Sources

Outside of the LOD world we can find sources of information that, even though they do not use RDF as their information representation format, enable us to access information in a easily automatable way. We are referring to Web sources that expose information that is encoded with formats such as JSON and XML. Those are formats that encode pure–pristine–information, not mixing it with presentation markup. This makes the extraction process from these sources easier than from those in, for instance, HTML.

The main difference of these sources from those belonging to the LOD initiative is that the intended meaning of the information exposed at the remote sites is not automatically translatable to the meaning at the local side[4]. In addition to that, RDF triples are always RDF triples, whereas two different websites may use different strategies to encode their data, even though both use XML as the format for data representation.

In summary, highly-structured sources—using our terminology—enable us to easily retrieve information, but a dedicated processing is necessary for each and every one source in order to extract that information.

C. Semi-structured Sources

We call semi-structured sources to those that encode information in such a way that it is mixed with presentation markup. The most common of its kind are sources that expose information that is represented in HTML—that is to say, traditional Web pages. In order to retrieve and extract information from a source of this type, we need to browse the DOM tree of the page and identify concrete DOM elements that represent particular pieces of information. That is, we need to kind of reverse-engineering the web page in order to figure out the mappings between presentation-related tags and their meanings.

The good news is that the DOM tree is usually nothing else that a blur reflection of the pure information stored in a relational database[5], after having been intermingled with presentation markup. Thus, once we get rid of all the presentation tags, we can get the most precious pieces of information[6].

In summary, information retrieval and extraction from semi-structured sources is more costly that from highly-structured ones, because in addition to a dedicated processing for each and every source we also need to reverse engineer and get rid of a lot of uninteresting pieces of data.

[3]RDF is particularly well suited to represent information in knowledge management applications, as it is able to represent may different and, at times abstract, concepts.

[4]Conversely, LOD conventions allow for sharing information together with its meaning.

[5]According to [6], most Web sites use a SQL database as their persistence solution.

[6]That is the alegory of data mining, to get "gold" from heaps of "dust".

III. Application in the iTEC Project

iTEC promotes an educational practice in which students interact in small projects which include participation in events, speeches with experts, and all that seasoned with the use of technology. The *leiv motiv* of this new paradigm is "Resources beyond Content". Thus, in iTEC educational resources go beyond educational content, frequently consumed in the form of textbooks, and they include technological resources, as well as cultural events and people who are experts on some knowledge area.

The initial assumption was that teachers were going to have a hard time trying to figure out which events could have the greatest relevance among such an enormous offer. This hypothesis was the main motivation for the design, development, and posterior rolling out of the SDE (Scenario Development Environment) [7], [8], which is a software system that works as a recommender, in such as way that a certain teacher, during the phase of preparing an educational experience, may rely on the SDE for selecting the most interesting events to be attended by learners during the performance of the educational experience. The SDE's recommendation algorithm has several factors into account, such as the appropriateness of the event and the proximity of the event to the school [9].

After three years of project and more than 2000 pilots in schools across Europe, that starting hypothesis has shown to be wrong. That is to say, the number of events registered at the P&E directory is not of the scale that it was supposed to be, and thus it does not makes necessary to use a recommender. The main reason for the low number of registered events is that the registering process is very time consuming, in addition to being very error-prone. In order to tackle this drawback, it was implemented and integrated in the SDE an enrichment module, which automatically extracts and processes huge amounts of data coming from relevant Web sites that list applications, events, and experts in different areas of knowledge. Figure 2 shows events that were extracted from the Web in the SDE user interface.

A. Sources of information

All across the Internet there are a great number of websites that offer information on applications, events, and experts. On applications we extract information from three websites that we find particularly interesting:

- Softonic[7] is a huge repository of information on standalone applications. From every entry on that repository we can extract information such as its description, the operating system required, tags, and what is very important: the rating from users.
- Softpedia[8] is, similarly to softonic, a big database of information about applications.
- AlternativeTo[9] is an incredible resource for extracting meaning about the features of an application. For each application, AlternativeTo provides a list of "substitutes"—applications with similar functionalities and that may suit a similar need. This is extraordinarily useful from the point of view of populating a database aimed at serving as the knowledge base of a recommender, because we can extract very relevant information that may lead to more accurate recommendations. As an example, AlternativeTo enables us to extract information such as: Skype is similar to Google Hangouts, and Firefox is similar to Chrome.

Regarding experts, two websites are the most relevant ones:

- Google Scholar[10] is an enormous database of researchers. From each registry we can extract very useful information, such as the name and position of researchers. To extract the information on their location—which in the context of iTEC is a very important piece of information, because experts of a near location should gain relevance in an hypothetical recommendation—we use an mechanism that takes as an input the position of the researcher (which is a text string that usually contains their affiliation) and their email address. The email address, in most cases, enables us to get the IP address of the expert's institution—provided that the institutions hosts the mail server, which is often true. The affiliation can be processed with NLP software[11] to get rid of the position and retrieve the institution, and then geocode the institution.
- LinkedIn[12] is a huge social network targeted at professionals. Unlike Google Scholar, which sorts entries by relevance[13], in LinkedIn we cannot measure the relevance of a particular professional. In order to overcome that difficulty—it is neither reasonable nor efficient to replicate all the LinkedIn public entries—we rely on Google's relevance calculations. Let's see that with an example, if we want to find experts on, let's say, biology we look for "biology site:www.linkedin.com" in Google. In this way, we get a bunch of search results ordered by the relevance that Google assigns to those entries.

In order to get events we need to follow a brute-force approach[14]. All across Europe there are a great number of

[7]http://www.softonic.com
[8]http://www.softpedia.com
[9]http://alternativeto.net
[10]http://www.scholar.google.com
[11]To that end we use the Geocoder library, which is available for the Ruby programming language.
[12]www.linkedin.com
[13]Google Scholar measures relevance as the number of references that researchers gathered to all their publications.
[14]At the time of writing, we extracted information from the following websites: www.spainisculture.com, www.discoveringfinland.com, www.unesco.org, www.finnbay.com, www.openeducationeuropa.eu, www.visitportugal.com, www.ulisboa.pt, www.uio.no (the University of Oslo), visit-hungary.com, visitbudapest.travel, visitbrussels.be, www.belgica-turismo.es, www.ualg.pt (University of Algalve), noticias.up.pt (University of Porto), www.globaleventslist.elsevier.com (worldwide conference registry).

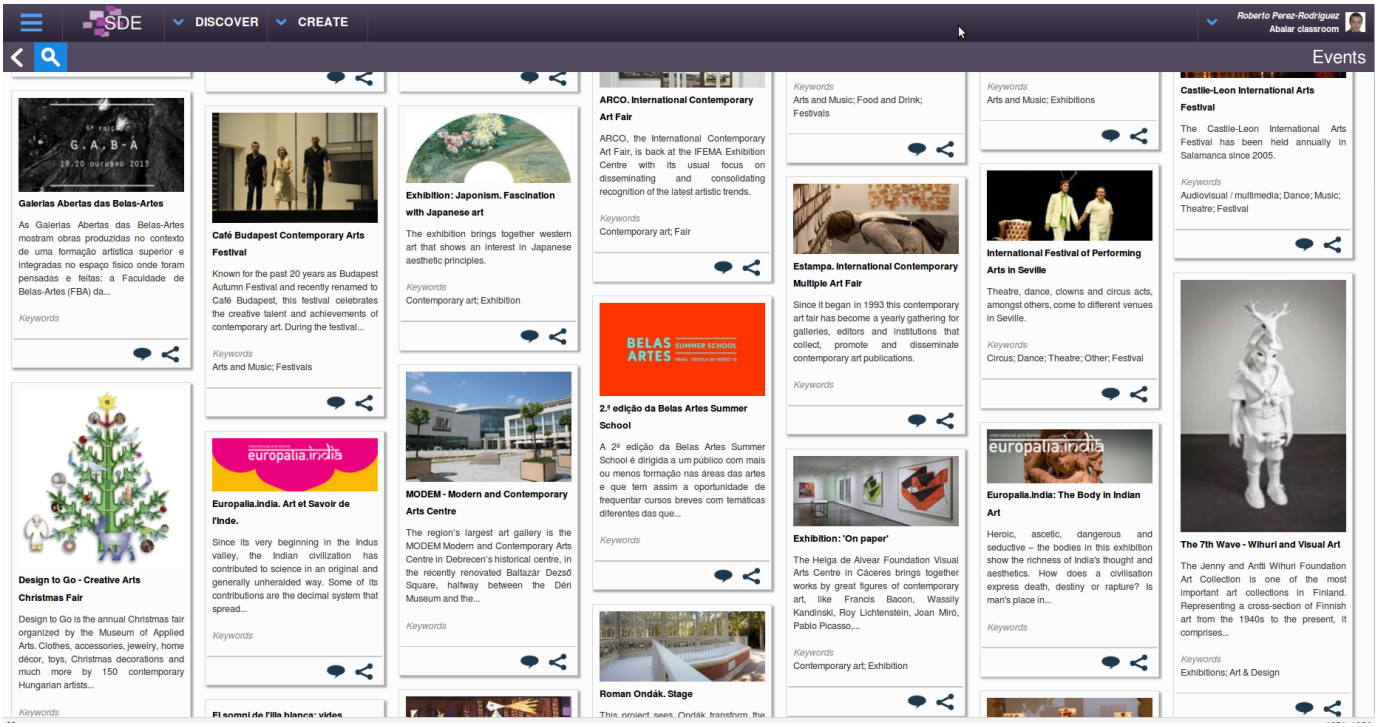


Fig. 1. Search results for events on "art" in the SDE user interface

websites that offer information on cultural events. Some of them, such as the Open Education Europa[15] initiative, are at a European scale. Conversely, others such as the Spain is Culture[16] portal are at a lesser scale.

B. Considerations on the Information Extraction Technique

There are several factors that you need to take into account in order to make an efficient scraping. The first one is that the scraping process of a web site may take quite a long time, so it should be performed as a background process. Besides, you have to take care of not launching too many requests to a given web server in a short time interval, because that could be seen as a Denial of Service attack, and the web server could block your IP.

Taking all the above into consideration, the scraping process starts with studying the HTML layout of the involved web pages. In order to do that efficiently we use the Firefox Firebug extension. The list of events in the Spain is Culture portal serves an an example of our procedure. We use Firebug for inspecting the HTML structure of the document.The next step is to test CSS selectors using the JavaScript console that is embedded in Firebug, issuing some JavaScript sentences that use jQuery.

Once we identify the concrete DOM elements that contain relevant information[17] we are ready to write some Ruby code that automates the parsing of applications, events, or experts. To that end, we make uses of the Nokogiri[18] library. Nokogiri is an open source HTML, XML, SAX, and Reader parser.

If we are dealing with events, once we got the venue of an event we try to get its coordinates. To that end, we make use of the Geocoder[19] gem, which issues requests to the Google Maps API. Thus, we geolocate all the events. The following snippet of code shows how we parse the text that represents the address and, from that, we get the latitude and longitude of the event using Geocoder:

```
1 address = event.css('a')[0]['href'].split("/")[3]
2 coordinates = Geocoder.coordinates(address)
3 latitude = coordinates[0]
4 longitude = coordinates[1]
```

[15]The Open Education Europa portal was launched by the European Commission as part of the Opening Up Education initiative. Its main purpose is to provide a focal access point to European OER in different languages. These OER are targeted to researchers, teachers, and students. According to information provided by the portal, it has more than 38000 registered users and receives more than 55,000 monthly visits.

[16]The Spain is Culture portal is devoted to promote and disseminate Spanish culture. The website presents the most outstanding of Spanish culture, and it includes a wide range of cultural offers, such as monuments, artistic styles, artists and places of interest. The navigation through the portal allows for discovering cultural events going from several starting points. You can search by a localisation, a town or autonomous region; by the historic period or artistic style; and, besides these ones, you can also search by the type of target audience.

[17]We identify which element in the DOM tree contains the information on the venue of a certain event in the Spain is Culture portal. The same matching has to be made for each field in the detailed view—the name, the description, the date range and so on. To ease that process, we can use the JavaScript console to test CSS selectors in jQuery until we find the correct one.

[18]http://nokogiri.org/

[19]www.rubygeocoder.com

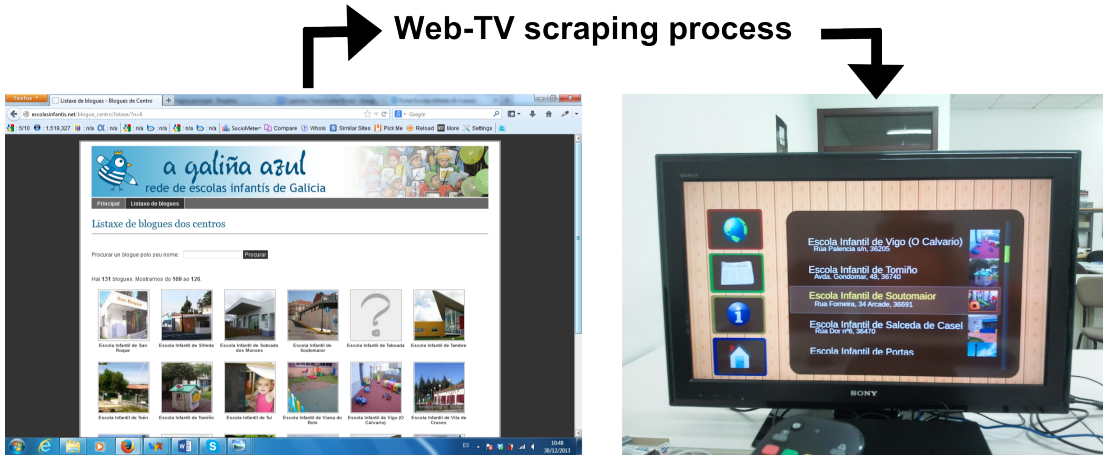


Fig. 2. The contents displayed on the TV screen are taken from the Web portal.

One important thing to consider when extracting events is to check if the scraped event is a new event, an update of an existing event, or a duplicate. To that end, we build a hash with the fields that make up an event. If the event is already present in the database but the calculated hash is different, then the update the fields of the event. Conversely, if the event is already present and the hash is identical, then we discard the scraped event because it is a duplicate. The following snippet of code shows how to create a hash with some of the fields that make up an event:

```
hash_event = Digest::MD5.hexdigest(name + description
    + start_date + end_date + latitude.to_s + longitude.
    to_s)
```

The filtering of duplicates is of special importance when dealing with events. Many times, when an event is published for the first time some of the fields might not be set yet. For instance, the final date might not be known yet; or maybe the venue is not fixed yet.

IV. Information Extraction for Interactive Digital TV Applications

Visualisation of content on digital TV depends on how digital information is presented—the same is true on smartphones and tablets. As it can be observed in smartphones, it is increasingly frequent to develop native applications so that the user does not have to access contents on the cloud through a Web browser. Following the same approach, in order to show third-party content on a digital TV we need either accessing an API that offers information in a structured way or accessing directly to content in HTML format, received from HTTP requests; HTML content is then processed in order to obtain and structure the information that will be later shown in the TV screen.

Regarding the way of obtaining the information that will be displayed we can distinguish between two different models for obtaining information: a passive one and an active one. Those depend on the needs of the final system that we are developing for. In the passive model, the user tells the system which information he/she wants to visualise at the time. The application is thus, at that very moment, in charge of obtaining the information requested, processing and displaying it in the TV screen—following the specifications for that information. On the contrary, in the active model, the application is continously retrieving information and processing it so that it may be ready when the user demands to see it.

Following is the description of two applications that were built on environments for digital TV, each one making use of a different model for obtaining information.

A. *BerceTV*

This native application enables parents of children in early childhood education to access the information that is published in the Galician Portal of Early Education Schools[20] though their digital TVs. Thus, parents are able to access both information of their children as well as diverse public information that is shown in the portal.

The application acts as a middleware between the TV and the Web portal. In this way, every action the user performs is converted—in real time—into an HTTP request that is sent to the portal. The information thus received is processed, filtered, and structured so that it may be displayed in an interface adapted to the characteristics of a TV screen. Figure 2 illustrates this process.

Since all the information is obtained in real time, this can result on requesting redundant information such as images. In order to improve the performance and efficiency, the system includes a cache layer in charge of receiving and managing multimedia information. This operating mode prevents downloading duplicated information; besides, it keeps the response time low, because the obtaining of multimedia information is performed in independent threads, which avoid

[20]http://escolasinfantis.net

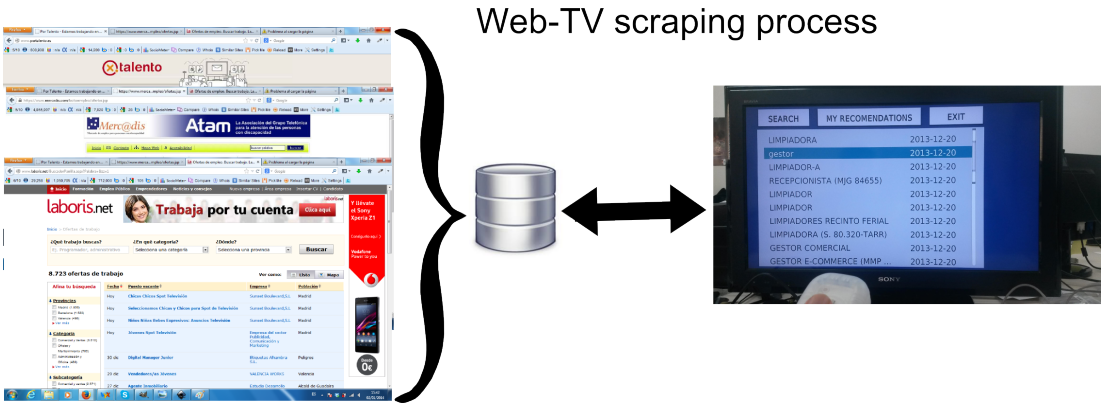


Fig. 3. Job offers are gathered from different sources and then displayed on the TV screen.

blocking the normal flow of the program in await for some information.

Besides obtaining public information that everybody can access to, one of the main features of this application is to let parents to access private information about their children. In order to access that information, every parent has a username and a password that enables them to enter the private area of the portal. The middleware uses those credentials to emulate the behaviour of a human person and access to the private information requested by the user. Thus, parents can see in their TV screen diverse information about their children, which is uploaded by educators.

B. Emprego

This application enables users to access information on job offers right from their TVs. One of the objectives of this application is to facilitate job search and, therefore, the application integrates tools for filtering search results in accordance with users' preferences. Figure 3 illustrates how information extraction is used in the context of this project.

In order to being able to apply filters on a heterogeneus set of job offers that were published in different Web portals, it is necessary to perform a homogenization task in advance, which entails that job offers must be available before users perform any searches. To that end, the active model for obtaining information is used. Thus, the system gathers information about job offers in an autonomous way. That is to say, the insertion, modification, and deletion of job offers is performed in accordance with information that is retrieved periodically from target sources.

The heterogeneity of information coming from different sources makes necessary to define a model of basic job offer, which is then enriched with the information obtained. All the information gathered is stored in a central database that is accessed by users through a Web Services API. As the system finds and updates information, the entries in the database are modified so that they may be acessed by digital TV clients.

V. Conclusions and Future Work

This paper showed how different information extraction techniques were used in the context of several research projects.

In the context of the iTEC project, information extraction is crucial for having a database of technologies, events, and experts that is big enough to serve as the universe of things to recommend to teachers. The strategy based on information extraction avoids us having to delegate on personnel for entering information about technologies, events, and experts. In a similar way, in the context of the Emprego project, information extraction allows us to delegate on automated software tasks the work of entering information about job offers.

In the context of the Berce TV project, information extraction is used to retrieve information from the Web portal. That information is shown to the user in a way that results suitable for being displayed in a digital TV. This strategy based on information extraction allows for integrating a legacy system in a fully-functional way. In spite of not being the most elegant solution on paper, it is an efficient one in practice, because it allows to add a new user interface based on digital TV without having to tweak legacy code—which would entail a new concession contract.

Currently, we are working on a research line that has as its objective to automatically annotation and categorise information extracted from the Web.

Acknowledgements

The work presented in this paper was partially supported by the European Regional Development Fund (ERDF); the Galician Regional Government under agreement for funding the Atlantic Research Center for Information and Communication Technologies (AtlanTIC); the Spanish Government and the European Regional Development Fund (ERDF) under project TACTICA; the European Commission's FP7 programme – project iTEC: innovative Technologies for an Engaging Classroom (Grant no. 257566); and the Spanish Ministry of Science and Innovation under grant "Methodologies,

Architectures and Standards for adaptive and accessible e-learning (Adapt2Learn)" (TIN2010-21735-C02-01). The content of this paper is the sole responsibility of its authors and it does not represent the opinion of the European Commission, or the Spanish Ministry of Science and Innovation, which are not responsible of any use that might be made of the information contained herein.

REFERENCES

[1] C. Bizer, J. Lehmann, G. Kovilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "CBpedia - A crystallization point for the web of data," *Journal of Web Semantics*, vol. 7, no. 3, pp. 154–165, 2009.

[2] C. Chang, M. Kayed, M. Girgis, and K. Shaalan, "A survey on web information extraction systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1411–1428, 2006.

[3] E. Prud'hommeaux and A. Seaborne, "SPARQL Query Language for RDF," World Wide Web Consortium, W3C Recommendation, Tech. Rep., 2008.

[4] G. Tummarello, R. Delbru, and E. Oren, "Sindice.com: weaving the open linked data," in *6th International Semantic Web Conference*, 2007.

[5] A. Harth, A. Hogan, R. Delbru, J. Umbrich, S. O'Riain, and S. Decker, "SWSE: Answers Before Links!" in *6th International Semantic Web Conference*, 2007.

[6] B. He, M. Patel, Z. Zhang, and K. Chang, "Accessing the deep web," *Communications of the ACM*, vol. 50, no. 5, pp. 94–101, 2007.

[7] A. Canas Rodriguez, V. M. Alonso Roris, J. M. Santos Gago, L. E. Anido Rifon, and M. J. Fernandez Iglesias, "The iTEC-SDE recommendation algorithms," *International Journal of Systems and Control*, pp. 1–8, 2013.

[8] A. Cañas Rodríguez, V. M. Alonso Rorís, J. M. Santos Gago, L. E. Anido Rifón, and M. J. Fernandez Iglesias, "Providing event recommendations in educational scenarios," in *Management Intelligent Systems*. Springer, 2013, pp. 91–98.

[9] L. Anido, M. Caeiro, A. Cañas, M. Fernández, V. Alonso, and J. M. Santos, "ITEC - WP 10 D10.3. Support for implementing ITEC engaging scenarios V3," The European ITEC project homepage, EUN Partnership AISBL, Rue de Trèves 61, B-1040 Brussels, Tech. Rep., 2013. [Online]. Available: http://itec.eun.org/c/document_library/get_file?uuid=1f1576cf-96b6-46bb-b34b-f66eca0f3cdf&groupId=10136

Computing Polynomial Segmentation through Radial Surface Representation

Leticia Flores-Pulido, Gustavo Rodríguez-Gómez, Oleg Starostenko, Vicente Alarcón, and Alberto Portilla

***Abstract*—The Visual Information Retrieval (VIR) area requires robust implementations achieved trough mathematical representations for images or data sets. The implementation of a mathematical modeling goes from the corpus image selection, an appropriate descriptor method, a segmentation approach and the similarity metric implementation whose are treated as VIR elements. The goal of this research is to found an appropriate modeling to explain how its items can be represented to achieve a better performance in VIR implementations. A direct method is tested with a subspace arrangement approach. The General Principal Component Analysis (GPCA) is modified inside its segmentation process. Initially, a corpus data sample is tested, the descriptor of RGB colors is implemented to obtain a three dimensional description of image data. Then a selection of radial basis function is achieved to improve the similarity metric implemented. It is concluded that a better performance can be achieved applying powerful extraction methods in visual image retrieval (VIR) designs based in a mathematical formulation. The results lead to design VIR systems with high level of performance based in radial basis functions and polynomial segmentations for handling data sets.**

***Index Terms*—Subspace arrangement, data modeling, segmentation, polynomial function, radial basis surface representation.**

I. Introduction

THIS paper presents an improvement in content-based applications where visual information retrieval area requires formalization methods obtaining higher implementation performance. With approximately 100 % of retrieval cases, there is not yet a methodological formalism to design visual information retrieval systems. Implementations of visual information retrieval (VIR) systems imply items as a feature extraction method, a segmentation technique, and a similarity metric. Those items have no formal or mathematical implementation framework. The formal implementation can achieve an ideal design that assures a high performance a priori. No mathematical framework has been introduced into VIR systems design, because differences between methods and features of the image collections, as well as relative lack of implementation standards and applications requirements. Today, the last implementations in segmentation approaches have to become effectives and VIR items can be attached with mathematical formalism allowed by segmentation techniques and algebraic methods to data manipulation [4]. The efficiency between image retrieval metrics and organization data methods has not been formalized [20]. With the intention to formalize the design of VIR applications and to increasing performance into the design process, we have tested approximately one thousand of images from four different libraries used for retrieving tasks matching user queries versus image collections, modeling data distribution with radial basis function, testing five different approaches for polynomial representations, the segmentation approach responsible to cluster images in groups are classified by GPCA-MVT (robust generalized principal component analysis with multivariate timing) algorithm. GPCA-MVT is selected from three varieties of it. This paper uses 477 image data classified into tree subsets with a tridimensional representation from a polynomial function computed by thin plate spline radial basis function. The surface representation improves the GPCA-MVT with the substitution of radial basis and the substitution of data collection. The final result allows proposing a mathematical expression with the possibility of to measure the performance of VIR systems from its implementation phase.

Manuscript received on December 17, 2012, revised version on June 6, 2014, accepted for publication on June 12, 2014.

Leticia Flores-Pulido and Alberto Portilla are with PCSCE, Universidad Autónoma de Tlaxcala, Tlaxcala, Mexico (e-mail: {leticia.florespo, alberto.portilla}@udlap.mx).

Gustavo Rodríguez-Gómez is with Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, Mexico (e-mail: grodrig@inaoep.mx).

Oleg Starostenko and Vicente Alarcón are with Escuela de Ingeniería, Universidad de las Américas Puebla, Mexico (e-mail: {oleg.starostenko, vicente.alarcon}@udlap.mx).

A. Mathematical Formalization Problem

The rapidly increasing power of computers has led to the development of novel applications such as multimedia, image and video databases [1]. These applications take advantage of the increasing processing power and storage of computers, which rapidly process large amounts of data. The challenge has now become one of developing suitable tools and methods for manipulation of the available data. Given the enormous amount of the information contained in a multimedia data stream, it is reasonable to consider that a deeper comprehension of the data stream may need to be achieved through the integration of independent analysis of different aspects [2], [3]. VIR techniques are an astringent need to solve several problems related to the management of these multimedia data stream. There exist several studies dealing with visual information retrieval system such as color, texture or shape [21]. The content-based visual information retrieval systems can to obtain features describing the image in more detail.

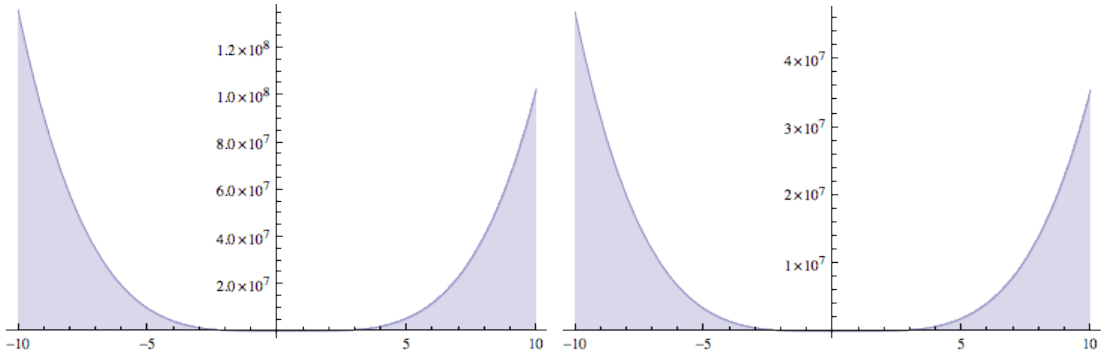


Fig. 1. Polynomial representation with (x, y, z) = (Red, Green, Zero) and (x, y, z) = (Red, Blue, Zero)

However, no standard has yet been formally established about:

- VIR's elements structure
- Relations between VIR's elements
- Restrictions of VIR's elements

The objective of modeling is to increase performance for VIR systems based on content-based image retrieval applications with mathematical abstraction. Practical implications referring to precision, recall and other performance evaluation metrics remain as an open problem in the field of visual information retrieval [21]. Each application content-based leads to its own ad-hoc designs, and it is valuable to know other factors involved in the process. The theoretical value of this model contributes to the mathematical handling of computer sciences within the visual information retrieval area. The VIR elements model is helpful to define constraints, parameters, standards and relations which could be helpful for content-based applications, in addition, it can be a support for systems which include high values in precision and recall [20]. Mathematical treatment implies definition of geometric dispersion and statistical analysis providing an abstract representation of data sets and improving techniques and classification results. The typical learning is statistical or probabilistic learning. Huge collections handle mixed data that are typically modeled as a group of samples $\{s_1, s_2, ..., s_n\} \in \Re$ is obtained from a learning approach with some probabilistic distribution. Each of the samples has a domain of values and they are composed of a set of vectors. Every vector can be modeled as a singular value array that describes relevant features of a sample as an image or an image collection a stellar spectrum or a document set. This approach allows retrieving images from a VIR system. With former fundamentals, this research proposes:

To design a mathematical abstraction for VIR systems design.

- To perform an analysis of a polynomial function as the representative model of data sets.
- To examine several radial basis functions as similarity metric.
- To use and modify the GPCA algorithm in basis computation.
- Finally, a mathematical formalization is proposed to encourage VIR systems efficient designs.

B. Organization of this paper

In this paper, we review the solutions to Problem 1.1 under no standards formalized yet. As a result, the scope of subjects to be covered ranging from VIR systems, from feature descriptors, and from segmentations with real data of images. Nevertheless, we hope to convince the reader that these subjects are strongly related one to another and they are crucial for researchers who want to gain a deep and complete formalization about the problem. The paper is organized as follows: Section 2 reviews the basic properties of formal polynomial representation. Section 3, we made some variation that allows us to estimate the segmentation in subspaces from sample points. Section 4, explains the mathematical formalization for VIR systems. A formula that can compute or increase the performance for content-based applications, especially for image retrieval is provided in section 5, and one explanation of how this formalization can be applied to several real-world applications is provided.

C. Polynomial Representation by Radial Basis Surface

The polynomial functions tested were useful for selection of dimensions of coil data. It is important to remember that a color descriptor was used to describe the image collection by its placement in the space. The performance of functions in two dimension or a polynomial representation and surface representation is exposed in Figure 1 and 2.

Figure 1 shows that the order of assignation (x, y) = (Red, Green) or (x, y) = (Green, Blue) is indistinct because finally, the representation is tridimensional, but a change in scales is observed for image collection polynomial representation. The coefficients are the parameter that will be useful for retrieval similarity metric.

Figure 2 indicates that surface representation is invariant to the order to assignation for (x, y) = (Red, Green) or (x, y) = (Green, Blue). The importance here is the variation

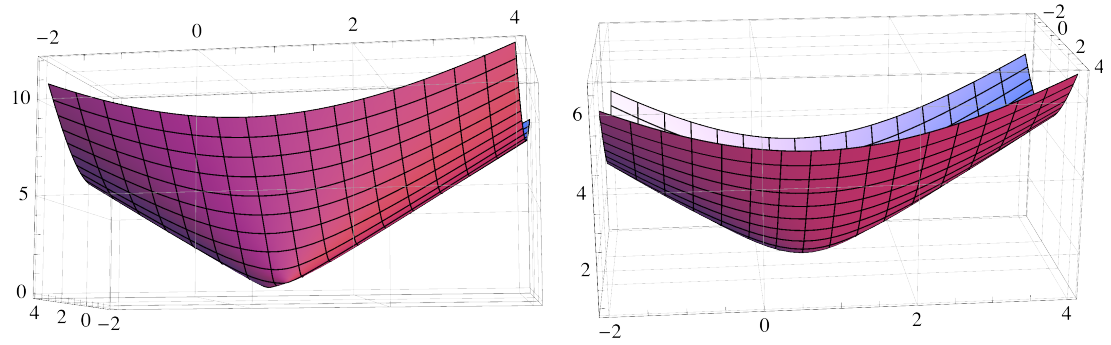


Fig. 2. Surface representation with $(x, y, z) =$ (Red, Zero, Blue) and with $(x, y, z) =$ (Red, Green, Zero)

of coefficients for similarity measure that are computed ad hoc to the algebraic distribution of the images or samples considered. Again, the scales of the surfaces do not change in a relevant difference. Finally, the Figure 3 implies that does not matters if assignation take care of (x, y) coordinates or if (x, z) coordinates are considered from collection data, but the only variation are the coefficients that are computed only from a representation in two dimensions.

D. Fundamentals and functions

Several works about subspace arrangements such as segmentation method in data sets have been developed. Subspace arrangements can be computed through Vanishing Ideals in polynomial rings. Some of the works detailing this process are [5], [6], and [7]. The concept of outliers in subspaces arrangements refers to elements that lie out of some data group well classified. Some papers help to detect and to avoid outliers in subspace arrangements such as [8], [10] and [11]. A variation for model detection of subspaces is described in [12], [13] and [14] as an alternative to subspace arrangements. Relevant related work deal with at GPCA (generalized components analysis) improvements as mentioned in [4], [15], [16], [17], and [18]. Recent methods and variations in linear hybrid modeling as mathematical treatments are showed in this section.

A relevant work for this research is [4], which is about a linear hybrid model based on subspace arrangements and Hilbert functions. The work [6] provides details about Hilbert function and vanishing ideal combination. The paper [15] explains details and considerations about subspace arrangements. The work [18] proposes an algebraic and geometric approach for subspace arrangements in a generalized principal component analysis. The main inspiration for this research was Yi Ma's work [4], and it is adapted to understand the mathematical formalization for VIR systems that pursue this research. A special variation of GPCA algorithm used to modeling data sets is presented in Section 1) below. The core of the similarity metric based in radial basis surface representation is explained in Section 2) below.

1) The Algorithm and the similarity metric: The original GPCA algorithm can be reviewed in [7] and the improved version in [4] for reader convenience.

2) Variation 1: Radial basis function: Given n distinct points $x_1, ..., x_n \in \Re^D$ where the function values $f(x_1, x_2, ..., x_n)$ are known $\Re^D$ are real values with dimension D, and we use an interpolate of the form:

$$s_n(x) = \sum_{i=1}^{n} \lambda_i \phi(\| x - x_i \|) + p(x), x \in \Re^D, \qquad (1)$$

where $\| . \|$ is the Euclidean norm. $\lambda_i \in \Re^D$ for $i = 1, ..., n$, with $p \in \Pi_n^d p$ (the linear space of polynomials in d variables of degree less than or equal to m), and Φ is a real valued that can take many forms. The most suitable function for this work is $\Phi(r) = r^2 log r, r > 0$ and $\Phi(0) = 0$ called surface splines [19], [20].

Let Φ be the function from surface splines:

$$\Phi(r) = \left\{ \begin{array}{ccc} r^k & k \in N, & k\ odd \\ r^k log(r) & k \in N, & k\ even \end{array} \right\} r \geq 0, \qquad (2)$$

where w is a coefficient empirically selected. Let m be any integer such that $m \geq m_\phi - 1$, and let Ω be a subset of N^d. Then we define $\mathcal{F}_{\Phi,m}(\Omega)$ and $\mathcal{A}_{\Phi,m}(\Omega)$ to be the linear function spaces. Let s and u be any functions in $\mathcal{A}_{\Phi,m}(\Omega)$. The semi-inner product is the expression:

$$\langle s, u \rangle = ((e-1)^{m_\phi}), \sum_{i=1}^{N(s)} \lambda u(y_i), \qquad (3)$$

where N is the total number of samples, lambda is the coefficients, and y_i is the output of Φ function. Thus Equation (6) is required as a semi-inner product that induces the semi-norm $\| . \| := \langle ., . \rangle^{1/2}$. The following section shows two variations of our approach. The first one is based in a basis computation of the image collection. The second variation is based in COIL-100 image collection as input data along with the RBF computation of the first variation.

3) Variation 2: Radial Basis Function as input parameter: The analysis of test for this step involves in the first place knowing the function that reveals the coefficients computed

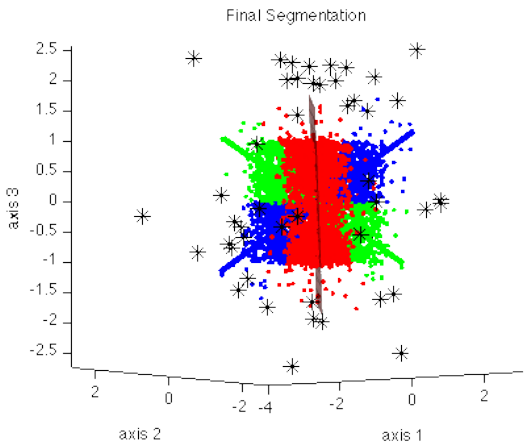


Fig. 3. Test 4 with COIL-100 as input data to RGPCA-MVT algorithm for segmentation data (see Table I)

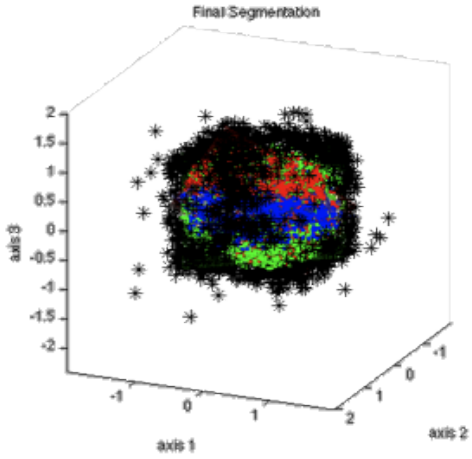


Fig. 4. The Test 12 with RBF-Basis as input data to RGPCA-MVT algorithm for segmentation data (see Table II)

for the polynomial function obtained from the best radial basis function computed as follows:

$$\varphi(x,y) = (x^2+y^2)log^{10}(\sqrt{x^2+y^2}) \quad (4)$$

The best RBF obtained was Spline, therefore, the polynomial function obtained as similarity metric measure for the first, second and third basis group for our research are computed using:

$$\begin{pmatrix} -21.3037\varphi[(x_{Q1}-x_1),(y_{Q1}-y_1)] \\ +1.58945\varphi[(x_{Q1}-x_2),(y_{Q1}-y_2)] \\ -5.39250\varphi[(x_{Q1}-x_3),(y_{Q1}-y_3)] \end{pmatrix}, \quad (5)$$

$$\begin{pmatrix} -21.3037\varphi[(x_{Q2}-x_1),(y_{Q2}-y_1)] \\ +1.58945\varphi[(x_{Q2}-x_2),(y_{Q2}-y_2)] \\ -5.39250\varphi[(x_{Q2}-x_3),(y_{Q2}-y_3)] \end{pmatrix}, \quad (6)$$

$$\begin{pmatrix} -21.3037\varphi[(x_{Q3}-x_1),(y_{Q3}-y_1)] \\ +1.58945\varphi[(x_{Q3}-x_2),(y_{Q3}-y_2)] \\ -5.39250\varphi[(x_{Q3}-x_3),(y_{Q3}-y_3)] \end{pmatrix}. \quad (7)$$

The next step implies to compute the basis for each group that is provided as input to the RGPCA-MVT algorithm. Those coefficients are computed by trial and error in the state of art, but in this case, the values are precise and established for each group of images from a random image of the group. This random image does not require be representative from the collection and does not require to be specially treated as in other works of the area. The basis for the three groups previously computed is exposed in Table I. These values are now provided as input to the RGPA-MVT algorithm. The Dimension 3 in Table I is not used for the SDM research objectives. The results obtained from RBF basis as input to the RGPCA-MVT algorithm are detailed in Table I. These results show that Average Basis Error increases, but the percentage of Error Classification is of 62.40 % (see Test 9). The classification accuracy of the segmentation obtained is 37.6 % that reflects 1 % of improvement of the RGPCA-MVT algorithm. The visual result of basis computation variation can be seen in Figure 4.

TABLE I
VALUES OF RADIAL BASIS FUNCTION PROVIDED AS BASIS INPUT FOR RGPCA-MVT ALGORITHM

Group	Dimension 1	Dimension 2	Dimension 3
1	0.131370	-0.056482	0.272775
2	0.555409	-0.002310	0.174136
3	0.749958	-0.057221	0.001160

The detected rejection rate obtained from the nine test from Table II is important only because is a warranty that at least one model is found for modeling the data collection of images, in contrast with Test 1, 2, 3, 5, 6, and 8, where no model was achieved for the segmentation data values.

TABLE II
VARIATION 2: MODIFICATION AT RGCPA-MVT ALGORITHM WITH RBF BASIS AS INPUT

Outlier Percentage	Dimension Groups	Detected Rejection	Segmentation Error	Average Basis
0.1600	(1,1,1)	No model	66.07%	81.17%
0.0800	(1,1,1)	No model	69.40%	68.60%
0.5000	(1,1,1)	No model	76.78%	65.64%
0.9000	(1,1,1)	20	66.56%	71.84%
0.1600	(2,1,1)	No model	65.31%	81.29%
0.0800	(2,1,1)	No model	68.69%	79.25%
0.5000	(2,1,1)	10	64.48%	89.70%
0.9000	(2,1,1)	No model	67.50%	76.26%
0.1600	(2,2,2)	46	62.40%	90.00%
0.0800	(2,2,2)	12	64.25%	90.00%
0.5000	(2,2,2)	35	63.26%	90.00%
0.9000	(2,2,2)	33	64.68%	90.00%

E. Mathematical Formalization with RGPCA-MVT

The main motivation of Segmentation Data Modeling (SDM) research for VIR Systems was generated starting on the following approaches: How can a mathematical model

of a visual information retrieval system be constructed? The kind of modeling for visual image retrieval systems imply the previous organization of the images, along with an important and powerful algorithm like RGPCA-MVT that has been successfully used in tracking techniques [4]. This algorithm allows a segmentation of groups applying a mathematical approach to establish a plane for groups of two dimensions or more and ensures a properly image retrieval regardless disadvantages of descriptor selected. How can at least one of visual information retrieval elements be constrained? The main visual information retrieval elements are: the image collection, a feature extractor or descriptor, a similarity measure, and an organization or classification approach. Fortunately, is possible to constrain the similarity measure by radial basis function specifically by Thin Plate Splines.

Another element that can be constrained for visual image retrieval systems is the method of classification or organization of the images, and this can be constrained by the RGPCA-MVT. The other element that can be constrained is that of the image collection which must be appropriate for the application. The other item that can be constrained in a certain manner is the image descriptor. For an image descriptor it is important to mention that if the image descriptor are well selected, the classification and the retrieval can be improved. It is important to test in further work with another texture approaches. How can performance in a VIR system through a mathematical point of view of its data set be increased? The most important point of view to increase the performance in a visual information retrieval system is the requirement of a careful selection of the Descriptor containing the main feature vector extraction of the data set. How can be helpful the prediction in new data set properties for VIR systems? The data set properties are inherently algebraic, always are present in data sets and can be modeled when a feature extractor method is applied.

F. Conclusions and Perspectives

The purpose of our modeling is to increase performance for VIR systems and we found the following answers at our questions research in section 4 leading to conclude about content-based image retrieval design through a mathematical modeling. The varieties of applications in the VIR area are evaluated with certain performance metrics as recall or precision just to mention a few. The challenge of each application implies to increment of certain metrics at 100%.

Most applications never achieve 100% in recall or precision. The initial attribution were as consequence to the user subjectivity to make some kind of classifications, but now we know that there are other elements implicated that could be systematically controlled in the data sets of this kind of systems (VIR or CBIR applications) as Segmentation method, Classification percentage and Similarity metric used. Our contribution provides a framework to detect certain aspects in VIR system designs to increase percentages of performance before they are implemented or further developed.

ACKNOWLEDGMENT

The authors would like to thank UAT, PROMEP, CONACYT, UDLAP, and INAOE facilities.

REFERENCES

[1] S. Little and S. Rueger, "Conservation of effort in feature selection for image annotation," in *IEEE Workshop on Multimedia Signals Processing, (MMSP 2009)*, Rio de Janeiro, Brazil, 2009, pp. 5–7.

[2] C. D. Manning, P. Raghavan, and H. Schütze, *An Introduction to Information Retrieval*. Cambridge, England: Cambridge University Press, 2009.

[3] K. Shearer, S. Venkatesh, and H. Bunke, "Rapid similarity retrieval from image and video," in *Multimedia Image and Video Processing*, L. Guan, S. Y. Kung, and J. Larsen, Eds. CRC Press, 2001, ch. 15, pp. 437–466.

[4] M. Y. Yang, H. Derksen, and R. Fossum, "Estimation of subspace arrangements with applications in modeling and segmenting mixed data," *SIAM Review*, vol. 50, no. 3, pp. 413–458, August 2008.

[5] A. Björner, I. Peeva, and J. Sidman, "Subspace arrangements defined by products of linear forms," *J. London Math. Soc.*, vol. 2, no. 71, pp. 273–288, 2005.

[6] H. Derksen, "Series of subspace arrangements," preprint, 2005. [Online]. Available: http://arxiv.org/abs/math/0510584

[7] R. Vidal and J. Piazzi, "A new GPCA algorithm for clustering subspaces by fitting," in *Differentiating and Dividing Polynomials, CVPR*, 2004, pp. 510–514.

[8] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, pp. 1289–1306, April 2006.

[9] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. WILEY, 2001.

[10] Y. Sugaya and K. Kanatani, "Outlier removal for motion tracking by subspace separation," *IEICE Trans. Inform*, vol. E86-D, pp. 1095–1102, 2003.

[11] A. Yang, S. Rao, and Y. Ma, "Robust statistical estimation and segmentation of multiple subspaces," in *Workshop on 25 years of RANSAC, IEEE International Conference on Computer Vision and Pattern Recognition*, 2006, pp. 99–106.

[12] K. Kanatani, "Motion segmentation by subspace separation and model selection," in *The 8th International Conference in Computer Vision*, vol. 2, Vancouver, Canada, July 2001, pp. 586–591.

[13] ——, "Geometric inference from images, what kind of statistical model is necessary?" *Systems and Computers in Japan*, vol. 35, no. 6, pp. 1–9, 2006.

[14] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[15] K. Huang, A. Wagner, and Y. Ma, "Identification of hybrid linear time-invariant systems via subspace embedding and segmentation," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 3, 2004, pp. 3227–3234.

[16] Y. Ma and R. Vidal, "A closed form solution to the identification of hybrid ARX models via the identification of algebraic varieties," in *Proceedings of the International Conference on Hybrid Systems Computation and Control*, 2005, pp. 449–465.

[17] S. Rao, A. Yang, A. Wagner, and Y. Ma, "Segmentation of hybrid motions via hybrid quadratic surface analysis," in *Proceedings of IEEE International Conference on Computer Vision*, 2005, pp. 2–9.

[18] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Trans. Pattern Anal. Machine Intelligence*, vol. 27, pp. 1–15, 2003.

[19] L. Flores-Pulido, *Data Segmentation Modeling for Visual Information Retrieval Systems*. Puebla, Mexico: Universidad de las Américas, 2011.

[20] L. Flores-Pulido, W. E. Estrada-Cruz, and J. A. Chávez-Aragón, "An image retrieval system based on feature extraction for machine vision using three similarity metrics," in *Congreso Internacional Cars Fof 2007, 23rd ISPE International Conference on Cad/Cam, Robotics Factories of the Future*, Universidad Militar De Nueva Granada, Bogotá, Colombia, August 2007.

[21] A. Iske, *Multiresolution Methods in Scattered Data Modeling*, ser. Lecture Notes in Computational Science and Engineering. Springer, 2004, vol. 37.

Mejoramiento de la consistencia entre la sintaxis textual y gráfica del lenguaje de *Semat*

Carlos Mario Zapata Jaramillo, Rafael Esteban Arango Sanchez, Leidy Diana Jiménez Pinzón

***Resumen*—Semat (Software Engineering Method and Theory) es una iniciativa que permite representar prácticas comunes de metodologías ya existentes mediante los elementos de su núcleo, los cuales se describen en términos de un lenguaje. Este lenguaje tiene una sintaxis gráfica y una textual. La sintaxis textual se describe mediante el metalenguaje EBNF (Extended Backus-Naur Form) que se utiliza como notación de gramáticas de libre contexto para describir un lenguaje formal. Sin embargo, la sintaxis textual de los elementos del núcleo en algunos casos presenta inconsistencia con la sintaxis gráfica. Por ello, en este artículo se propone la modificación del lenguaje textual mediante un análisis gramatical al lenguaje de Semat con el fin de lograr una relación consistente entre la sintaxis textual y gráfica de los elementos del núcleo de Semat.**

***Palabras clave*—Análisis gramatical, EBNF, Semat, sintaxis textual.**

Improving the Consistency between Textual and Graphical Syntax of the Language of *Semat*

***Abstract*—Semat (Software Engineering Method and Theory) is an initiative that allows representing common practices of existing methodologies by its core elements, which are described in terms of a language. This language has a graphical and a textual syntax. The textual syntax is described using meta-language EBNF (Extended Backus-Naur Form), which is used as context-free grammar notation to describe a formal language. However, the textual syntax of core elements in some cases is inconsistent with the graphical syntax. Therefore, in this paper we propose a modification of textual language by parsing the language of Semat in order to achieve a consistent relationship between textual and graphical syntax of the core elements of *Semat*.**

***Keywords*—Parsing, EBNF, Semat, textual syntax.**

I. Introducción

Semat es una iniciativa que apoya un proceso para redefinir la ingeniería de software con base en una teoría sólida, principios probados y mejores prácticas. A diferencia de otros intentos para crear una teoría general de la ingeniería de software, en Semat se generaliza la ingeniería de software identificando acciones y elementos universales, que se describen mediante un lenguaje sencillo y universal que permite la descripción de las prácticas comunes de metodologías existentes y así lograr que se puedan evaluar, comparar y medir [1]. Su núcleo incluye un grupo de elementos esenciales que son universales para todo esfuerzo de desarrollo de software y extensibles para usos específicos, lo que permite asumir que Semat no se resiste ante nuevas ideas, ya que cualquier metodología se puede representar mediante sus elementos en el núcleo [2].

El lenguaje de Semat posee una sintaxis abstracta, la cual se compone de una sintaxis textual y una sintaxis gráfica. La sintaxis gráfica comprende la representación, una forma visual, de los elementos del núcleo de Semat, mientras que la sintaxis textual, se encuentra descrita en el metalenguaje EBNF (*Extended Backus-Naur Form*), presenta una descripción formal de cada uno de los elementos del núcleo [3].

La notación Backus-Naur (BNF por sus siglas en inglés) se creó inicialmente para describir la sintaxis del lenguaje de programación ALGOL 60 y se utiliza desde entonces como notación para las gramáticas libres de contexto, las cuales permiten describir la estructura sintáctica de muchos (aunque no todos) lenguajes [4].

Tal como se puede ver en la fig. 1, una gramática consta de un conjunto de no-terminales, terminales y una serie de reglas de producción. Un no-terminal se define en una regla de producción, mientras que un terminal es un símbolo del lenguaje que se está definiendo. En una regla de producción, el no-terminal (que aparece en la parte izquierda) se define en términos de una secuencia de símbolos no-terminales y terminales (que se encuentran en la parte derecha) [5].

```
<símbolo> ::= <expresión con símbolos>
```

Figura 1. Expresión BNF

EBNF es un conjunto de expansiones de BNF, por lo cual presenta pequeñas diferencias sintácticas y algunas operaciones adicionales. En ella se incorporan algunos conceptos de la notación sintáctica de Wirth con el propósito de definir la gramática de los lenguajes de programación (lenguajes formales) [6].

En este artículo se propone una revisión de la sintaxis textual de los elementos del núcleo de Semat, con el fin de encontrar las inconsistencias existentes entre las representaciones gráficas de los diferentes elementos y las especificaciones descritas en la sintaxis textual. Existen elementos gráficos a los cuales les faltan conexiones que se definen en el lenguaje textual y existen expresiones que representan de forma incompleta lo que el lenguaje gráfico muestra. Adicionalmente, estos elementos se

Manuscrito recibido el 19 de abril de 2014; aceptado para la publicación el 17 de junio del 2014.

Todos los autores están con la Universidad Nacional de Colombia, sede Medellín, Colombia (correos: {cmzapata, raearangosa, ldjimenezp}@unal.edu.co).

TABLA I.
OPERADORES EBNF

Operador	Descripción
‘ ’	Delimitador de carácter. Por ejemplo: **‘b’**
“ ”	Delimitador de cadenas de caracteres. Por ejemplo: **“>=”**
|	Alternativa. Por ejemplo: **‘b’** | **‘a’**
( )	Delimitadores de agrupamiento. Por ejemplo: **(‘a’ | ‘e’ | ‘i’ | ‘o’ | ‘u’)**
. .	Rango. Por ejemplo: **‘1’. .‘7’**
?	Opcional (cero o una vez). Por ejemplo: **(‘0’. .‘9’)?**
*	Repetición cero o más veces. Por ejemplo: **(‘0’. .‘9’)***
+	Repetición una o más veces. Por ejemplo: **(‘0’. .‘9’)+**
~	Negación. Por ejemplo: **~(‘b’ | ‘a’)**

especifican con invariantes y algunas operaciones adicionales definidas en OCL (*Object Constraint Language*). Sin embargo, las expresiones en OCL de Semat aún presentan problemas de consistencia que se deberían corregir para conseguir un uso adecuado de los elementos del lenguaje y poder representar las prácticas y los métodos. Por las razones anteriores, se toman algunos ejemplos específicos del *Essence* [3] y se realiza la especificación correcta de acuerdo con la sintaxis gráfica propuesta.

Este artículo se organiza de la siguiente manera: en la sección II se presenta el marco teórico que incluye una descripción de Semat, los elementos del núcleo y su lenguaje propio y la descripción de la sintaxis del metalenguaje EBNF; en la sección III se presentan los antecedentes sobre las especificaciones con la sintaxis textual que tiene Semat en algunas de sus representaciones gráficas; en la sección IV se propone una modificación a la especificación de dichos ejemplos tomados del *Essence* y, por último, en la sección V se concluye y se propone el trabajo futuro.

II. MARCO TEÓRICO

A. EBNF *(Extended Backus-Naur Form)*

Una especificación EBNF es un sistema de reglas donde existe sólo una acción primitiva. Es una ecuación sintáctica que permite definir una categoría sintáctica S, mediante una expresión E [6], como se aprecia en la fig. 2. La secuencia “::=” es el metasímbolo de la producción. Una producción se puede considerar como la definición de S en términos de E. <S> es no-terminal y la <E> consiste en una lista de términos sintácticos alternativos que incluyen caracteres y operadores.

```
<S> ::= <E>
```

Figura 2. Especificación EBNF

En la notación EBNF se encuentran los operadores que se definen en la Tabla I [6].

Los términos en una ecuación sintáctica se separan con operadores. Tal como se puede ver en la fig. 3, la expresión E sólo se puede reemplazar con uno y solo un término T, pues el metasímbolo | es una “o” excluyente [6].

$$\langle E \rangle ::= \langle T_1 \rangle \mid \langle T_2 \rangle \mid \ldots \mid \langle T_n \rangle, \quad n > 0$$

Figura 3. Términos sintácticos alternativos

Tal como se puede ver en la fig. 4, cada término T se puede reemplazar con la concatenación de factores [6].

$$\langle T \rangle ::= \langle F_1 \rangle \mid \langle F_2 \rangle \mid \ldots \mid \langle F_n \rangle, \quad n > 0$$

Figura 4. Factores sintácticos

Según se aprecia en la fig. 5, a un factor también se le pueden asignar operadores como: Opción (a), Repetición de cero o más veces (b) y Repetición por lo menos una vez (c) [6].

```
<T> ::= [E]    <T> ::= {E}    <F> ::= [E*]
```

(a) (b) (c)

Figura 5. Opción y Repetición

Una forma de verificación es la construcción del árbol de derivación, el cual permite representar gráficamente la especificación EBNF, planteada desde el elemento inicial (raíz) hasta los elementos terminales (hojas) [6]. Un árbol de derivación que ejemplifica la especificación EBNF se muestra en la fig. 6. La forma gráfica se muestra en la fig. 7.

```
<entero con signo> ::= <signo><entero>
<signo> ::= '+' | '-'
<entero> ::= <dígito><entero> | <dígito>
<dígito> ::= '0' | '1'
```

Figura 6. Ejemplo de especificación EBNF (la tercera regla es recursiva)

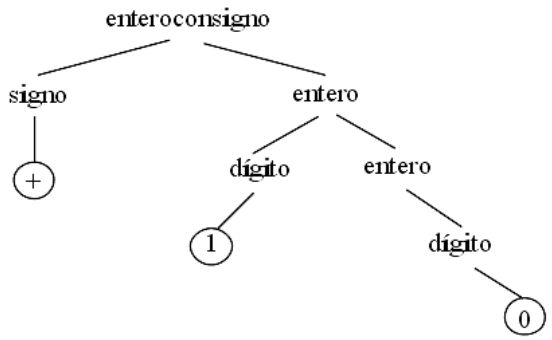


Figura 7. Ejemplo de árbol de derivación

B. *Semat (Software Engineering Method and Theory)*

La especificación del lenguaje de Semat se constituye utilizando una combinación de tres técnicas diferentes: un metamodelo, un lenguaje formal y lenguaje natural. El metamodelo expresa la sintaxis abstracta y algunas limitaciones

estructurales de las relaciones entre los elementos del núcleo. A raíz de la existencia de una invariante por cada elemento y las limitaciones, se proporcionan reglas de formación del idioma (semántica estática). Los invariantes y algunas operaciones adicionales se establecen utilizando el lenguaje OCL como lenguaje formal. La descripción de los elementos del núcleo y la semántica dinámica se describen mediante lenguaje natural acompañado de definiciones formales utilizando VDM (*Vienna Development Method*) [3].

La sintaxis textual del lenguaje del núcleo de Semat se especifica en EBNF. Además de los operadores descritos anteriormente de esta notación, se identifican dos elementos más [3]:

- ID: palabra específica que representa un identificador para un elemento definido.
- Ref: denota un símbolo que representa un identificador de un elemento (es decir, no del elemento definido).

La sintaxis gráfica del lenguaje de Semat proporciona una forma visual para cada uno de sus elementos, donde cada uno de estos elementos corresponde a un aspecto específico de un núcleo o método. En *Essence* [3] se dividen los elementos de Semat en diferentes categorías. En la fig. 8 se presentan las representaciones gráficas de los elementos descritos a continuación [3].

Grupos de elementos:

- Núcleo: conjunto de elementos usados para formar una base que describe la ingeniería de software.
- Método: es un conjunto de prácticas que forma la descripción de los esfuerzos que se realizan en una empresa. Los esfuerzos se visualizan mediante instancias como alfas, productos de trabajo, actividades y similares.
- Práctica: es un esfuerzo que se puede repetir y se realiza con un propósito específico.

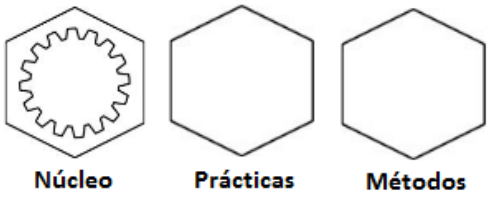


Figura 8. Sintaxis gráfica del grupo de elementos [3]

Elementos del núcleo: en la fig. 9 se presentan las representaciones gráficas de los elementos descritos a continuación [3].

- Alfa: se caracteriza por un conjunto de estados que representan su progreso y salud. Cada estado tiene una lista de chequeo que especifica los criterios necesarios para alcanzar un estado en particular.
- Estado: expresa una situación donde algunas condiciones se proponen.
- Lista de chequeo: lista que se necesita verificar en un estado.
- Espacios de actividad: son las "cosas que siempre hacemos" en ingeniería de software. Agrupan conjuntos de actividades que se realizan mientras se desarrolla un producto de software.
- Competencia: una característica del interesado o equipo que refleja la habilidad de hacer un trabajo. Se puede detallar en diferentes "niveles de competencia".

Figura 9. Sintaxis gráfica de los elementos del núcleo [3]

Elementos de las prácticas: En la fig. 10 se presentan las representaciones gráficas de los elementos descritos a continuación [3].

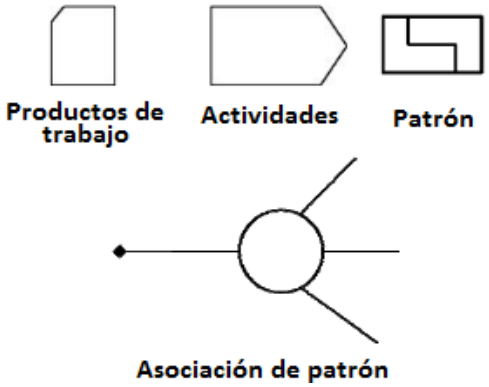


Figura 10. Sintaxis gráfica de los elementos de Semat [3]

- Productos de trabajo: artefactos que se utilizan o se generan en una práctica.
- Actividad: se define como una o más clases de puntos de trabajo y la guía sobre cómo realizarlas.
- Patrón: es una descripción de una estructura en una práctica.
- Asociación de patrón: permite conectar un patrón con otros elementos en una práctica.

III. Antecedentes

En esta Sección se presenta la sintaxis textual de elementos de Semat y la comparación con el lenguaje gráfico. Se propone el estudio de sólo algunos elementos para profundizar acerca de

las ambigüedades encontradas entre los dos tipos de sintaxis que componen su lenguaje.

En el lenguaje textual de Semat se usan tres símbolos globales para la declaración de los elementos:

- ID: se utiliza como identificador del elemento definido y es una palabra única que se utiliza, también, como nombre del elemento.
- Ref, este símbolo denota una referencia a un elemento definido y es la unión del ID con la palabra "Ref" al final.
- STRING, aunque es un tipo de dato; si se usa después del ID se utiliza como una descripción del elemento.

A. *Root Elements*

Dentro de la clasificación de elementos de Semat, existe un grupo denominado "Root elements", que contiene la especificación de los elementos descritos en la Sección anterior y, adicionalmente, propone la organización de otros elementos del lenguaje textual de Semat.

GroupElement:
Kernel | Practice | Library | PracticeAsset | Method;

De acuerdo con la expresión anterior cualquier *Kernel*, *Practice*, *Library*, *PracticeAsset* o *Method* es un *GroupElement*.

A partir del razonamiento de la primera definición, se presentan las siguientes especificaciones que completan el grupo "*Root Elements*":

PatternElement:
Alpha | AlphaAssociation | AlphaContainment |WorkProduct |WorkProductManifest | Activity | ActivitySpace | ActivityAssociation | Competency | Pattern;

Algunos de estos elementos contienen categorías de elementos en su definición, por lo que la cantidad de elementos aumenta, así:

PracticeElement:
PatternElement | ExtensionElement | MergeResolution | UserDefinedType;

Cualquier *PatternElement, ExtensionElement, MergeResolution* o *UserDefinedType* es un *PracticeElement*, por lo que la definición es equivalente a la siguiente:

PracticeElement:
Alpha | AlphaAssociation | AlphaContainment | WorkProduct |WorkProductManifest | Activity | ActivitySpace | ActivityAssociation | Competency| Pattern | ExtensionElement | MergeResolution | UserDefinedType;

AnyElement:
GroupElement | PracticeElement | State | Level | CheckListItem | CompetencyLevel | PatternAssociation | Tag | Resource;

KernelElement:
Alpha | AlphaAssociation | AlphaContainment | ActivitySpace | Competency | Kernel | ExtensionElement | MergeResolution | UserDefinedType;

StateOrLevel:
State | Level;

AlphaOrWorkProduct:
Alpha | WorkProduct;

AbstractActivity:
Activity | ActivitySpace;

PracticeContent:
PracticeElement | Practice | PracticeAsset;

MethodContent:
Practice | ExtensionElement | MergeResolution;

B. *Element Groups*

En esta sección se profundiza sobre los elementos de esta categoría que presentan inconsistencia con el lenguaje gráfico.

Kernel:
'kernel' ID ':' STRING
('with rules' STRING)?
('owns' '{' KernelElement* '}')?
('uses' '{' KernelElementRef (',' KernelElementRef)* '}')?
(AddedTags)?;

La especificación del elemento *Kernel* incluye un ID y una descripción obligatoria del mismo. Adicionalmente, puede contener expresiones que lo definen como:

'with rules' es una descripción que contiene las posibles reglas de ese elemento.

'owns' en esta expresión se especifican los elementos (cero o muchos) que contiene el elemento *Kernel* que se está definiendo. Estos elementos pertenecen al grupo *KernelElement.*

'uses' expresión en la que se definen los posibles elementos (cero o muchos) que se relacionan con el elemento *kernel* que se está definiendo y estos elementos deben pertenecer al grupo *KernelElement*. El usar la expresión *KernelElementRef* implica que contiene el ID de los posibles (cero o muchos) *KernelElement.*

Practice:
'practice' ID ':' STRING
'with objective' STRING
('with measures' STRING(',' STRING)*)?
('with entry' STRING(',' STRING)*)?
('with result' STRING(',' STRING)*)?
('with rules' STRING)?
('owns' '{' PracticeElement* '}')?
('uses' '{' PracticeContentRef (','PracticeContentRef)*'}')?
(AddedTags)?;

La especificación del elemento *Practice* incluye un ID, una descripción y un objetivo obligatorios. Adicionalmente, puede contener expresiones que lo definen como:

'with measures' es una descripción que contiene las posibles medidas de ese elemento.

'with entry' es una descripción que contiene las posibles entradas de ese elemento.

'with result' es una descripción que contiene los posibles resultados de ese elemento.

'with rules' es una descripción que contiene las posibles reglas de ese elemento.

'owns' en esta expresión se especifican los elementos (cero o muchos) que contiene el elemento *Practice* que se está definiendo. Estos elementos pertenecen al grupo *PracticeElement.*

'uses' expresión en la que se definen los posibles elementos (cero o muchos) que se relacionan con el elemento *Practice* que se está definiendo y estos elementos deben pertenecer al grupo *PracticeContent*. El usar la expresión *PracticeContentRef* implica que contiene el ID de los posibles (cero o muchos) *PracticeContent.*

C. *Practice Elements*

Activity:
'activity' ID ':' STRING
(Resource(',' Resource)*)?
'targets' StateOrLevelRef (',' StateOrLevelRef)*
('with actions' Action (',' Action)*)?
('requires competency level' CompetencyLevelRef(',' CompetencyLevelRef)*)?
(AddedTags)?;

La especificación del elemento *Practice* incluye un ID, una descripción y el cumplimiento de al menos un objetivo (grupo *StateOrLevel*). Adicionalmente, puede contener expresiones que lo definen como:

Resource es una expresión que contiene una fuente.

'with actions' es una expresión que muestra las acciones que se tiene sobre alfas o productos de trabajo.

'requires competency level' es una expresión que muestra el ID de las competencias necesarias para la actividad que se está describiendo.

AddedTags es información adicional para describir la actividad.

D. *Auxiliary Elements*

Resource:
'resource' (UserDefinedTypeRef '=')? STRING;

Es una expresión que contiene una fuente o información de donde se puede conseguir información.

IV. Representación

Para este artículo se realiza la declaración del lenguaje textual de Semat de tal manera que el lenguaje textual sea consistente con el lenguaje gráfico. A continuación se profundiza sobre los elementos modificados:

A. *Resource*

Este elemento se cambió ya que la expresión *Resource* necesita sólo un STRING para la descripción de la fuente del elemento.

Resource:
'resource' ID ':' STRING;

Lenguaje gráfico:
No tiene

Conexión con otros elementos: **UserDefinedType**.

B. *Kernel, Practice y Activity*

Estos elementos se modificaron para que las representaciones realizadas con estas declaraciones no tengan asociaciones con elementos que no existen en el lenguaje gráfico.

Kernel

Dentro de la definición de este elemento se especifica que se compone y relaciona con *KernelElement*. Sin embargo en el lenguaje gráfico, el *Kernel* no tiene relaciones con otoros elementos.

Adicionalmente, en el lenguaje textual se encuentran inconsistencias. A partir de la definición de *Kernel* en BNF, se puede inferir que el *Kernel* tiene un *AlphaAssociation*. La inconsistencia surge cuando se conoce la definición de este último elemento, el cual sólo se usa entre Alphas.

AlphaAssociation:
Cardinality AlphaRef '--' STRING '-->' Cardinality AlphaRef (AddedTags)?;

Por lo tanto la especificación del elemento para lograr la consistencia es la siguiente:

Kernel:
'kernel' ID ':' STRING
('with rules' STRING)?
(AddedTags)?;

De esta manera solo se espera un ID obligatoriamente al definir un *Kernel* y puede o no tener una descripción con sus reglas y algunos comentarios adicionales sobre el elemento. De esta manera, esta especificación es consistente con lo que muestra el lenguaje gráfico (véase la fig. 11).

Lenguaje gráfico:

Conexión con otros elementos: Ninguna

Figura 11. Conexiones del Kernel en el lenguaje gráfico [6]

Practice

Dentro de la definición de este elemento se especifica que se compone de *PracticeElement* y relaciona con *PracticeContent*. Sin embargo, en el lenguaje gráfico, *Practice* no tiene relaciones con otros elementos.

Adicionalmente, en el lenguaje textual se encuentran inconsistencias. A partir de la definición de *Practice* en BNF, se puede inferir que *Practice* tiene *PracticeElement*. A su vez *PracticeElement* puede ser *PatternElement* y éste último puede ser *ActivityAssociation*. La inconsistencia surge cuando se conoce la definición de este último elemento, el cual sólo se usa entre *Activities*.

Adicionalmente, la definición de *PatternElement* no resulta ser congruente con el lenguaje gráfico al tener asociada una *ActivityAssociation*.

ActivityAssociation:
AbstractActivityRef '--' STRING '-->' AbstractActivityRef (AddedTags)?;

Por lo tanto la especificación del elemento para lograr la consistencia es la siguiente:

Practice:
'practice' ID ':' STRING
'with objective' STRING
('with measures' STRING(',' STRING)*)?
('with entry' STRING(',' STRING)*)?
('with result' STRING(',' STRING)*)?
('with rules' STRING)?
(AddedTags)?;

De esta manera, sólo se espera un ID y un objetivo obligatorios al definir *Practice* y puede o no tener una descripción con sus medidas, entradas, resultados y algunos comentarios adicionales sobre el elemento. Así, esta especificación es consistente con lo que muestra el lenguaje gráfico (véase la fig. 12).

Lenguaje gráfico:

Conexión con otros elementos: Ninguna

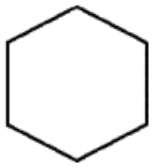


Figura 12. Conexiones de *Practice* en el lenguaje gráfico [3]

Activity

Dentro de la definición de este elemento se especifica que se compone de *StateOrLevelRef* o *CompetencyLevel.* Sin embargo, en el lenguaje gráfico, *Activity* no tiene relaciones con otros elementos.

Por lo tanto la especificación del elemento para lograr la consistencia es la siguiente:

Activity:
'activity' ID ':' STRING
(Resource(',' Resource)*)?
(AddedTags)?;

Por ello, sólo se espera un ID obligatorio al definir *Activity* y puede o no tener una descripción con la fuente y algunos comentarios adicionales sobre el elemento.

Las actividades, en consecuencia, se crean de forma independiente y para crear las diferentes asociaciones que muestra el lenguaje gráfico se utiliza el *ActivityAssociation*.

ActivityAssociation:
AbstractActivityRef '--' STRING '-->' AbstractActivityRef (AddedTags)?;

De esta manera esta especificación es consistente con lo que muestra el lenguaje gráfico (véanse las Fig. 13 y 14).

Lenguaje gráfico:

Conexión con otros elementos: **Activity** y **Activity Space**.

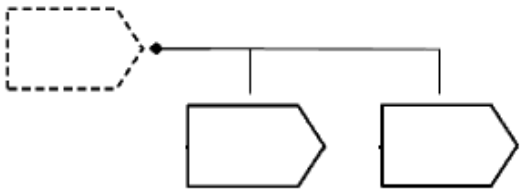


Figura 13. Conexión de **Activity** con **ActivitySpace** en el lenguaje gráfico [3]

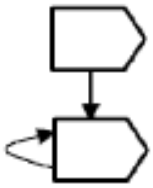


Figura 14. Conexión de **Activity** con **Activity** en el lenguaje gráfico [3]

C. *PatternElement*

En el lenguaje gráfico de Semat, el elemento *Pattern* tiene una asociación con cualquier elemento. Sin embargo, en el lenguaje textual aparecen más elementos auxiliares con los cuales el elemento *Pattern* no debería tener asociación, por ejemplo: **AlphaAssociation**, **AlphaContainment** y **ActivityAssociation**.

Por lo tanto, este elemento se editó con el fin de que los elementos que utilizan este objeto no puedan hacer referencia a las asociaciones entre los elementos que antes se presentaban.

PatternElement:
Alpha | WorkProduct | WorkProductManifest | Activity | ActivitySpace | Competency | Pattern;

De esta manera, esta especificación es consistente con lo que muestra el lenguaje gráfico (véase la fig. 15).

Lenguaje gráfico:

Conexión con otros elementos: Todos.

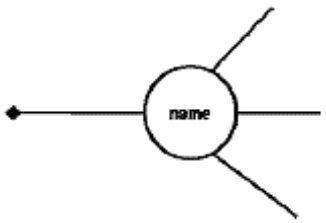


Figura 15. Conexión del **PatternElement** [3]

V. CONCLUSIONES

Las definiciones de un lenguaje dependen de tres componentes: semántica, sintaxis y pragmática. La semántica revela el significado de las frases que son correctamente formadas, la pragmática estudia los factores extralingüísticos del acto comunicativo que permiten entender, inferir o interpretar lo que el emisor de la frase quiso dar a entender y la sintaxis define la relación formal entre los constituyentes de un lenguaje, es decir, se refiere a la forma correcta de formar las frases en el lenguaje. Una definición formal de la sintaxis se denomina gramática, la cual se utiliza para especificar de manera finita el conjunto de cadenas de símbolos que constituyen un lenguaje.

La forma BNF es una manera explícita de representar la gramática libre de contexto y se utiliza en lenguajes de programación para hacer y expresar la estructura de manera formal mediante símbolos. EBNF permite el uso de algunas reglas adicionales que facilitan la especificación y completitud del lenguaje.

Aunque la declaración de Semat en el lenguaje EBNF tiene una sintaxis correcta, la semántica muestra un lenguaje diferente al lenguaje gráfico. Semat cuenta con dos tipos de lenguajes, el lenguaje gráfico y el lenguaje textual, los cuales describen los diferentes elementos que componen el núcleo de esta iniciativa. Sin embargo, los lenguajes aun presentan falencias, pues se encontraron inconsistencias en la especificación de los elementos. Los elementos auxiliares que define el núcleo de Semat en EBNF permiten la definición de otros elementos del núcleo.

En el lenguaje textual de Semat se especifican elementos auxiliares que son necesarios para poder representar las asociaciones entre los elementos del núcleo de Semat. Sin embargo, una especificación incorrecta de estos elementos puede representar asociaciones incorrectas, por ejemplo *PatternElement* el cual abarca elementos como *AlphaAssociation* y *ActivityAssociation* los cuales no son parte del elemento *Pattern*.

Siendo *Essence* una especificación aún en construcción en el OMG, se espera que sigan surgiendo versiones que contengan inconsistencias como las que se analizaron en este artículo. Por ello, el trabajo futuro se basa en la continuación del análisis de los elementos textuales de la especificación y su adaptación al lenguaje gráfico que describe Semat. Otra línea de trabajo futuro que da continuidad a este artículo se relaciona con otros elementos de la especificación de *Essence* que, aún sin declarar sus relaciones en el lenguaje textual o en el gráfico, permiten conexiones de tipo pragmático por medio de diagramas descriptivos o por uso de sus elementos.

AGRADECIMIENTOS

El proyecto identificado con el código 18907 y que lleva por título “Especificación formal en OCL de reglas de consistencia entre métodos de desarrollo basados en planes, representado en el núcleo de SEMAT”, que financia la Dirección de Investigación de la Sede Medellín (DIME), adscrita a la Universidad Nacional de Colombia suministró los fondos para la realización de este artículo.

REFERENCIAS

[1] I. Jacobson, P. P. W. Ng, P. E. McMahon, I. Spence, S. Lidman, C. M. Zapata (traductor), “La esencia de la ingeniería de software: El núcleo de Semat”, *Revista Latinoamericana de Ingeniería de Software*, vol. 3, pp. 71–78, 2013.

[2] N. Chomsky, “The independence of grammar”, en *Syntactics structures,* S. Wendland, Walter de Gruyter GmnH & Co. KG, Berlín, 1957, pp. 117.

[3] Essence – *Kernel and Language for Software Engineering Methods. Versión 1.3*, 2013.

[4] B. L. Kurtz, K. Slonneger, ”Specifying syntax”, en *Formal syntax and semantics of programming languages. A laboratory based approach*, 1995, pp. 625.

[5] D. E. Knuth, “Backus Normal Form vs. Bakus Naur Form”, *Communications of the ACM*, vol. 7, no. 12, 1964, pp. 735–736.

[6] L. Reynoso, M. Genero, M. Piattini, “Towards a metric suite for OCL Expressions expressed within UML/OCL models”, *Journal of Computer Science & Technology*, vol. 4, no.1, 2004, pp. 38.

Journal Information and Instructions for Authors

I. Journal Information

Polibits is a half-yearly open-access research journal published since 1989 by the *Centro de Innovación y Desarrollo Tecnológico en Cómputo* (CIDETEC: Center of Innovation and Technological Development in Computing) of the *Instituto Politécnico Nacional* (IPN: National Polytechnic Institute), Mexico City, Mexico.

The journal has double-blind review procedure. It publishes papers in English and Spanish (with abstract in English). Publication has no cost for the authors.

A. Main Topics of Interest

The journal publishes research papers in all areas of computer science and computer engineering, with emphasis on applied research. The main topics of interest include, but are not limited to, the following:

- Artificial Intelligence
- Natural Language Processing
- Fuzzy Logic
- Computer Vision
- Multiagent Systems
- Bioinformatics
- Neural Networks
- Evolutionary Algorithms
- Knowledge Representation
- Expert Systems
- Intelligent Interfaces
- Multimedia and Virtual Reality
- Machine Learning
- Pattern Recognition
- Intelligent Tutoring Systems
- Semantic Web
- Robotics
- Geo-processing
- Database Systems
- Data Mining
- Software Engineering
- Web Design
- Compilers
- Formal Languages
- Operating Systems
- Distributed Systems
- Parallelism
- Real Time Systems
- Algorithm Theory
- Scientific Computing
- High-Performance Computing
- Networks and Connectivity
- Cryptography
- Informatics Security
- Digital Systems Design
- Digital Signal Processing
- Control Systems
- Virtual Instrumentation
- Computer Architectures

B. Indexing

The journal is listed in the list of excellence of the CONACYT (Mexican Ministry of Science) and indexed in the following international indices: LatIndex, SciELO, Periódica, e-revistas, and Cabell's Directories.

There are currently only two Mexican computer science journals recognized by the CONACYT in its list of excellence, *Polibits* being one of them.

II. Instructions for Authors

A. Submission

Papers ready for peer review are received through the Web submission system on www.easychair.org/conferences/?conf=polibits1; see also updated information on the web page of the journal, www.cidetec.ipn.mx/polibits.

The papers can be written in English or Spanish. In case of Spanish, author names, abstract, and keywords must be provided in both Spanish and English; in recent issues of the journal you can find examples of how they are formatted.

Only full papers are reviewed; abstracts are not considered as submissions. The review procedure is double-blind. Therefore, papers should be submitted without names and affiliations of the authors and without any other data that reveal the authors' identity.

For review, a PDF file is to be submitted. In case of acceptance, the authors will need to upload the source code of the paper, either Microsoft Word or LaTeX with all supplementary files necessary for compilation. Upon acceptance notification the authors receive further instructions on uploading the camera-ready source files.

Papers can be submitted at any moment; if accepted, the paper will be scheduled for inclusion in one of forthcoming issues, according to availability and the size of backlog. While we make every reasonable effort for fast review and publication, we cannot guarantee any specific time for this.

B. Format

The papers should be submitted in the format of the IEEE Transactions 8x11 2-column format, see http://www.ieee.org/publications_standards/publications/authors/author_templates.html. (while the journal uses this format for submissions, it is in no way affiliated with, or endorsed by, IEEE). The actual publication format differs from the one mentioned above; the papers will be adjusted by the editorial team.

There is no specific page limit: we welcome both short and long papers, provided that the quality and novelty of the paper adequately justifies its length. Usually the papers are between 10 and 20 pages; much shorter papers often do not offer sufficient detail to justify publication.

The editors keep the right to copyedit or modify the format and style of the final version of the paper if necessary.