VIII Jornadas de Ciencia e Ingeniería de Servicios

Sistedes 2012

JISBD | PROLE | ACTAS JCIS

Sistedes
Sociedad de Ingeniería de Software y Tecnologías de Desarrollo de Software

IN LUMINE SAPIENTIA · UNIVERSITAS ALMERIENSIS

Almería, 17 al 19 de Septiembre

Editores: Mª Valeria de Castro | José Manuel Gómez | Luis Iribarne

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): Actas de las "VIII Jornadas de Ciencia e Ingeniería de Servicios (JCIS'2012)", Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

JCIS 2012

VIII Jornadas de Ciencia e Ingeniería de Servicios (JCIS)

Almería, 17 al 19 de Septiembre de 2012

Editores:
Ma. Valeria de Castro
José Manuel Gómez
Luis Iribarne

Actas de las "*VIII Jornadas de Ciencia e Ingeniería de Servicios (JCIS)*"
Almería, 17 al 19 de Septiembre de 2012
Editores: Ma. Valeria de Castro, José Manuel Gómez, Luis Iribarne
http://sistedes2012.ual.es
http://www.sistedes.es

ISBN: 978-84-15487-26-5
Depósito Legal: AL 672-2012

Prólogo

El presente volumen contiene los artículos seleccionados para presentación en las *VIII Jornadas de Ciencias e Ingeniería de Servicios* (JCIS 2012) celebrado en Septiembre de 2012 en Almería, España.

El principal objetivo de las Jornadas es proporcionar un foro de discusión e intercambio de conocimiento y experiencias en el ámbito de la Ciencia de Servicios. El interés no sólo se centra en los nuevos avances científicos, sino también en las tecnologías existentes en torno a la computación orientada a servicios y los procesos de negocio, las nuevas prácticas de ingeniería de servicios y las lecciones aprendidas por medio de experiencias reales. Las JCIS, que celebran este año su octava edición, son el resultado de la integración de las Jornadas Científico-Técnicas en Servicios Web y SOA (JSWEB) y el Taller sobre Procesos de Negocio e Ingeniería de Servicios (PNIS). Seguir contando después de ocho años, y en el contexto de crisis en el que nos encontramos actualmente, con un foro de encuentro que nos permita intercambiar conocimiento y experiencias entre grupos de investigación de distintas Universidades españolas y profesionales de la Administración Publica y de la Industria, es sin duda un triunfo de nuestra comunidad que debemos y queremos destacar.

En esta edición se han recibido 22 contribuciones para su revisión, de las cuales 4 eran artículos publicados ya previamente en congresos y revistas de reconocido prestigio. Todos las contribuciones fueron revisadas por, al menos, dos miembros del comité de programa. Como resultado de este proceso de revisión, se seleccionaron 14 trabajos largos para su presentación en las jornadas y otros 3 como trabajos cortos. Los trabajos han sido organizados en seis sesiones temáticas dobles que se presentarán a lo largo de dos días de jornadas de los siguientes temas: “SOA, Tecnologías para Servicios Web y Aplicaciones”, “Ingeniería de Servicios” y “Procesos de Negocio”.

Nos gustaría agradecer a todos aquellos que de un modo u otro han contribuido a la organización de estas Jornadas. En primer lugar, a todos los autores de los artículos enviados a JCIS 2012, y a los miembros del Comité de Programa por su disponibilidad y dedicación a la hora realizar las revisiones. Agradecer además a nuestros colaboradores: SRII (Service Research & Innovation Institute), ATI, Novática, INES y la Red Científico-Tecnológica en Ciencias de los Servicios financiada por el Ministerio de Economía y Competitividad. Finalmente, agradecemos a la Sociedad de Ingeniería del Software y Tecnologías de Desarrollo del Software (SISTEDES) y a la organización por parte de los miembros de la Universidad de Almería. Conocemos la dificultad de la organización de este tipo de eventos, y máxime en la situación de crisis económica en la que nos encontramos, por lo que destacamos y agradecemos enormemente vuestro esfuerzo y dedicación en la realización de estas Jornadas.

Gracias a todos y esperamos que disfrutéis de las Jornadas y de vuestra estancia en Almería.

Almería, Septiembre 2012
Ma. Valeria de Castro y José Manuel Gómez
Presidentes del Comité científico

Prologo de la Organización

Las jornadas SISTEDES 2012 son un evento científico-técnico nacional de ingeniería y tecnologías del software que se celebra este año en la Universidad de Almería durante los días 17, 18 y 19 de Septiembre de 2012, organizado por el Grupo de Investigación de Informática Aplicada (TIC-211). Las Jornadas SISTEDES 2012 están compuestas por las XVII Jornadas de Ingeniería del Software y de Bases de Datos (JISBD'2012), las XII Jornadas sobre Programación y Lenguajes (PROLE'2012), y la VIII Jornadas de Ciencia e Ingeniería de Servicios (JCIS'2012). Durante tres días, la Universidad de Almería alberga una de las reuniones científico-técnicas de informática más importantes de España, donde se exponen los trabajos de investigación más relevantes del panorama nacional en ingeniería y tecnología del software. Estos trabajos están auspiciados por importantes proyectos de investigación de Ciencia y Tecnología financiados por el Gobierno de España y Gobiernos Regionales, y por proyectos internacionales y proyectos I+D+i privados. Estos encuentros propician el intercambio de ideas entre investigadores procedentes de la universidad y de la empresa, permitiendo la difusión de las investigaciones más recientes en ingeniería y tecnología del software. Como en ediciones anteriores, estas jornadas están auspiciadas por la Asociación de Ingeniería del Software y Tecnologías de Desarrollo de Software (SISTEDES).

Agradecemos a nuestras entidades colaboradoras, Ministerio de Economía y Competitividad (MINECO), Junta de Andalucía, Diputación Provincial de Almería, Ayuntamiento de Almería, Vicerrectorado de Investigación, Vicerrectorado de Tecnologías de la Información (VTIC), Enseñanza Virtual (EVA), Escuela Superior de Ingeniería (ESI/EPS), Almerimatik, ICESA, Parque Científico-Tecnológico de Almería (PITA), IEEE España, Colegio de Ingenieros Informática de Andalucía, Fundación Mediterránea, y a la Universidad de Almería por el soporte facilitado. Asimismo a D. Félix Faura, Director de la Agencia Nacional de Evaluación y Prospectiva (ANEP) de la Secretaría de Estado de I+D+i, Ministerio de Economía y Competitividad, a D. Juan José Moreno, Catedrático de la Universidad Politécnica de Madrid, presidente de la Sociedad de Ingeniería y Tecnologías del Software (SISTEDES), a D. Francisco Ruiz, Catedrático de la Universidad de Castilla-La Mancha, y a D. Miguel Toro, Catedrático de la Universidad de Sevilla, por su participación en la mesa redonda "*La investigación científica informática en España y el año Turing*"; a Armando Fox de la Universidad de Berkley (EEUU) y a Maribel Fernández del King's College London (Reino Unido), como conferenciantes principales de las jornadas, y a los presidentes de las tres jornadas por facilitar la confección de un programa de *Actividades Turing*. Especial agradecimiento a los voluntarios de las jornadas SISTEDES 2012, estudiantes del Grado de Ingeniería Informática y del Postgrado de Doctorado de Informática de la Universidad de Almería, y a todo el equipo del Comité de Organización que han hecho posible con su trabajo la celebración de una nueva edición de las jornadas JISBD'2012, PROLE'2012 y JCIS'2012 (jornadas SISTEDES 2012) en la Universidad de Almería.

Luis Iribarne
Presidente del Comité de Organización
@sistedes2012{JISBD;PROLE;JCIS}

Comité Científico

Presidentes del Comité Científico:

Ma. Valeria De Castro (Universidad Rey Juan Carlos)
José Manuel Gómez (ISOCO)

Coordinador de artículos ya publicados:

Marcos López Sanz (Universidad Rey Juan Carlos)

Miembros del Comité Científico:

Antonio Ruiz Cortés (Universidad de Sevilla)
Antonio Vallecillo (Universidad de Málaga)
Carlos Bobed (Universidad de Zaragoza)
Carlos Rodríguez Fernández (Universidad Complutense Madrid)
Diego López (Red Española de I+D, RedIRIS)
Daniel González Morales (Ag. Canaria de Inv., Innovación y Soc. de la Inf.)
Enrique Beltrán (Software AG)
Esperanza Marcos (Universidad Rey Juan Carlos)
Félix García (Universidad de Castilla-La Mancha)
Francisco Almeida Rodríguez (Universidad de La Laguna)
Francisco Javier Fabra (Universidad de Zaragoza)
Francisco Ruiz (Universidad de Castilla-La Mancha)
Guadalupe Ortiz Bellot (Universidad de Cádiz)
Jaime Cid (Oracle)
Jesús Arias Fisteus (Universidad Carlos III de Madrid)
Jesús Gorroñogoitia (Atos Origin)
Joan Pastor (Universitat Oberta de Catalunya)
Jordi Marco (Universidad Politécnica de Cataluña)
José Emilio Labra (Universidad de Oviedo)
José Hilario Canós (Universidad Politécnica de Valencia)
José M. López Cobo (Playence KG)
José Raúl Romero (Universidad de Córdoba)
Juan De Lara (Universidad Autónoma de Madrid)
Juan Hernández (Universidad de Extremadura
Juan José Moreno Navarro (Universidad Politécnica de Madrid)
Juan Manuel Murillo (Universidad Extremadura)
Juan Pavón (Universidad Complutense de Madrid)
Leire Bastida (Tecnalia)
Manuel Lama (Universidad de Santiago de Compostela)
Manuel Resinas (Universidad de Sevilla)
Marcos López Sanz (Universidad Rey Juan Carlos)
María del Carmen Penadés (Universidad Politécnica de Valencia)
María-Ribera Sancho (Universidad Politécnica de Cataluña)
Marta Patiño (Universidad Politécnica de Madrid)
Martín Álvarez Espinar (W3C Spain)
Mercedes Ruiz (Universidad de Cádiz)
Óscar Corcho (Universidad Politécnica de Madrid)
Pedro Alvarez (Universidad de Zaragoza)
Pere Botella (Universidad Politécnica de Cataluña)
Rafael Corchuelo (Universidad de Sevilla)
Santi Ristol (Atos Origin)
Silvia Acuña (Universidad Autónoma de Madrid)
Vicente Pelechano (Universidad Politécnica de Valencia)
Víctor Acinas (Inabensa)
Víctor Ayllón (Novayre)

Comité de Organización

Presidente:

Luis Iribarne (Universidad de Almería)

Miembros:

Alfonso Bosch (Universidad de Almería)
Antonio Corral (Universidad de Almería)
Diego Rodríguez (Universidad de Almería)
Elisa Álvarez, Fundación Mediterránea
Javier Criado (Universidad de Almería)
Jesús Almendros (Universidad de Almería)
Jesús Vallecillos (Universidad de Almería)
Joaquín Alonso (Universidad de Almería)
José Andrés Asensio (Universidad de Almería)
José Antonio Piedra (Universidad de Almería)
José Francisco Sobrino (Universidad de Almería)
Juan Francisco Inglés (Universidad Politécnica de Cartagena)
Nicolás Padilla (Universidad de Almería)
Rosa Ayala (Universidad de Almería)
Saturnino Leguizamón (Universidad Tecnológica Nacional, Argentina)

Colaboradores JCIS

Índice de Contenidos

Chala Invitada

Sesión 1: SOA, Tecnologías para Servicios Web y Aplicaciones
Chair: Dr. José Manuel Gómez

Sesión 2: Ingeniería de Servicios
Chair: Dr. Pedro Alvarez

Sesión 3: Procesos de Negocio
Chair: Dr. Inmaculada Medina

Sesión 4: Ingeniería de Servicios II
Chair: Dr. Vicente Pelechano

Sesion 5: SOA, Tecnologías para Servicios Web y Aplicaciones II
Chair: Dr. Víctor Ayllón

Sesion 6: Procesos de Negocios II
Chair: Dr. Juan Manuel Vara

Sesión 1

SOA, Tecnologías para Servicios Web y Aplicaciones

Chair: *Dr. José Manuel Gómez*

Sesión 1: SOA, Tecnologías para Servicios Web y Aplicaciones
Chair: Dr. José Manuel Gómez

Ruediger Gad, Juan Boubeta-Puig, Martin Kappes and Inmaculada Medina-Bulo. *Leveraging EDA and CEP for Integrating Low-level Network Analysis Methods into Modern, Distributed IT Architectures*.

Sergio Hernández, Javier Fabra, Pedro Álvarez and Joaquín Ezpeleta. *Una solución SOA para ejecutar workflows científicos en entornos Grid heterogéneos*.

Ricardo Jiménez, Marta Patino and Iván Brondino. *CumuloNimbo: Una Plataforma como Servicio con Procesamiento Transaccional Altamente Escalable*..

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): Actas de las "VIII Jornadas de Ciencia e Ingeniería de Servicios (JCIS'2012)", Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

Leveraging EDA and CEP for Integrating Low-level Network Analysis Methods into Modern, Distributed IT Architectures

Rüdiger Gad[2], Juan Boubeta-Puig[1], Martin Kappes[2], and Inmaculada Medina-Bulo[1]

[1] Department of Computer Science and Engineering, University of Cádiz, Spain
{inmaculada.medina, juan.boubeta}@uca.es
[2] University of Applied Sciences Frankfurt am Main
Nibelungenplatz 1, 60318 Frankfurt am Main, Germany
{kappes, rgad}@fb2.fh-frankfurt.de

Abstract. Computer networks are crucial for the operation of Information Technology (IT) infrastructures. For assuring and maintaining the functionality of networks and with this of IT systems in general, accurate and up-to-date information about networks and the incidents in them is of vital importance. To allow a proper, accurate, and timely assessment this information must be efficiently communicated to the relevant analysis applications that rely on it. In this paper we propose an approach on obtaining and efficiently communicating information gathered with means of low-level network analysis methods from spatially distributed and heterogeneous data sources. Thereby, we leverage existing technologies from the fields of network analysis, Event-driven Architecture (EDA), and Complex Event Processing (CEP) and combine these into a novel distributed network analysis system approach that can be integrated into todays, modern, distributed IT architectures.

Keywords: network analysis, network surveillance, EDA, CEP

1 Introduction

Information Technology (IT), nowadays, is a fundamental part of many productive, organizational, and other kinds of environments. Many businesses, governments, or other application fields in general like health care or traffic control heavily rely on operational IT systems. Faults or non-availability of IT systems quickly result in serious consequences ranging from financial losses to potential hazards to human health.

Computer networks form the basis for nearly all IT infrastructures as used today. Problems in computer networks can quickly lead to faults in or non-availability of the corresponding IT systems. Thus, the functionality of IT systems highly depends on working computer networks.

Many factors can impact the proper functioning of computer networks and in consequence of IT systems. Such factors are, e. g., attacks, hardware failures, or configuration errors.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 13-26, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

In order to maintain operating and functional network infrastructures it is paramount that incidents possibly impacting the network functionality are detected, reported, and reacted on early, quickly, and properly. A comprehensive, detailed, and up-to-date overview of a network, its connections, and the entities in it like hosts or services is the basis on which decisions about changes etc. in networks are made.

Computer networks usually have large spatial and logical extent. Already Local Area Networks (LANs) may easily grow very large and complex. Networks of larger institutions not only consist of a single LAN but of several LANs, usually representing the individual facilities or sites, that are in turn connected to a bigger overall network via Wide Area Network (WAN) connections. In order to acquire a comprehensive overview of such networks data has to be collected at different places inside the network.

Furthermore, there exists a large range of different methods for gathering information about a network and the traffic in it [1,2,3,4]. Typically each method provides different data and fits a special purpose featuring certain benefits as well as drawbacks when compared to other methods. In order to offer an as detailed and as comprehensive data basis as possible ideally multiple different network analysis methods should be employed in a combined fashion in order to efficiently complement each other. However, existing implementations are often not designed to work in conjunction with each other and usually feature highly heterogeneous data and formats.

Additionally, networks are no static entities but change over time. As businesses grow or requirements change, networks change as well to adapt to these new situations. In order to maintain a high level of quality and coverage when gathering information about network and network traffic, the analysis and surveillance equipment should seamlessly change with the surveyed network. Thus, the analysis and surveillance approach must be flexible to change with the observed network, ideally at runtime.

Finally, it would be desirable if existing, state-of-the-art solutions could be integrated into existing higher-level IT infrastructures like service oriented architectures (SOA) [5,6]. By this collected data could be easily made available for a larger range of different applications. This way, by enabling different types of applications working on the same data, the available data could be used much more efficiently. Moreover, new types of applications could become possible that further improve the possibilities and ways of processing, analyzing, or displaying the data.

Many modern IT infrastructures, on the other hand, already employ Web Services (WS) or a form of SOA. Such high-level abstraction services and infrastructures typically aim at easing interoperability between different components, e. g., by providing abstractions for communication infrastructures, data exchange, or remote procedure calls. These systems typically provide general, unified, and standardized ways for interoperability. With these abstractions distributed as well as heterogeneous components can be effortlessly integrated into bigger systems. Thus, on the first glance, these approaches appear ideal for

achieving the goals of connecting and integrating different, spatially distributed, or heterogeneous data sources like those mentioned for network analysis and for communicating the resulting data.

However, we discovered that there are some important aspects that have to be taken into account. In the rest of this paper we research the possibilities, challenges, and caveats of integrating low-level network analysis methods into high-level infrastructures and where possible will show solutions for the problems we identified. Thereby, we leverage principles of Event-driven Architecture (EDA) [9] and Complex Event Processing (CEP) [10] with the goal to eventually integrate low-level network analysis methods with high-level infrastructure abstractions like Event-driven Service Oriented Architectures (SOA 2.0) [11].

The rest of the document is structured as follows: Firstly, we explain the background of the employed technologies. We keep the focus of this discussion the application field as targeted in this paper. Secondly, we outline our approach and discuss possible issues as well as ways for solving these. Afterwards, we present the implementation of our prototype. Subsequently, we discuss our findings and results. Finally, we conclude with summarizing our work and we give an outlook on future work we have planned in the context of the research as presented here.

2 Background

For the work presented in this paper technologies from different fields had been combined. These fields include, network analysis, EDA, and CEP. In the following we give a short, introductory explanation for each field as well as references to present research and implementations.

2.1 Network Analysis

In a nutshell, network analysis in the context of computer networks is the process of gathering and analyzing data about computer networks, the entities inside such networks, and the network traffic [1,3]. Techniques for network analysis and surveillance include low- and high-level, passive and active, or distributed and non-distributed approaches [2,4].

Network monitoring systems like Nagios [7] or Zabbix [8] already use distributed and heterogeneous data sources, but these systems usually employ higher-level data sources. These and other existing distributed approaches usually use their own protocols for communication instead of standardized infrastructures like SOA. Thus, these are more difficult to extend and also more difficult to integrate with other services and applications.

2.2 Event-driven Architecture

Event-driven Architecture (EDA) [9], in a nutshell, refers to a paradigm of computing and data processing in which the actual processing is not triggered by

timers, polling, synchronous method calls, or synchronous messages but asynchronously by events. Generally, EDA refers to all systems that use events as their main driving principle. Besides asynchronous communication EDA allows very loose coupling of the different components of a system. Thanks to this property very diverse components like spatially distributed as well as heterogeneous components can be effortlessly integrated.

Because events are usually sent and forwarded as they occur there is also the notion of an EDA having near real-time capabilities. Near real-time must not be confused with actual "real-time systems" as the constraints, e. g., with respect to timing etc. on real-time systems are much tougher.

2.3 Complex Event Processing

Essentially, Complex Event Processing (CEP) is a way for processing data in the form of events [10]. Within CEP systems events, the data, can be processed in different ways. Operations supported by CEP systems are, e. g., filtering, transforming, or correlation of events.

With CEP information can be very efficiently aggregated and concentrated to the most important meaning. This way, important information can be efficiently extracted and separated from unimportant information. Thus, CEP provides the big potential to efficiently eliminate unnecessary, "meaningless" data. This is important because of two reasons. Firstly, the processed and aggregated data is very likely much smaller than the original data, and thus it does not occupy as much space and requires only a fraction of the bandwidth for being communicated as compared to the original data. Secondly, analysts reading and interpreting the data are not flooded with unneeded information but can easily concentrate on the important parts.

Another property of CEP systems is that they, like EDA in general, ease the implementation of near real-time systems. Thereby, the meaning equals the near real-time property of EDA mentioned above. In fact, most CEP systems are implemented using an EDA.

There are already some approaches in the field of network analysis and surveillance that already employ CEP. The motivation for using CEP and event-driven principles is thereby similar to our motivation for using CEP in the work presented here, namely, the capabilities of CEP for easily integrating distributed sources as well as the powerful methods for deriving meaningful data from events. Examples of such work are intrusion detection [12,13] or the detection of distributed, concealed port scans across companies [14,15]. However, so far very low-level data acquisition as presented in this work has, to the best of our knowledge not yet been considered and researched.

3 Our Approach

The general goal of our approach is to integrate low-level network analysis methods into a high-level communication infrastructure with the intent of eventually

making the data gathered this way available via higher-level infrastructures like event-driven SOAs. The aim is to apply different, heterogeneous low-level network analysis methods and to design a system that can be easily extended to support more methods. The "sensors" that each utilize at least on of these methods for gathering the data are intended to be deployed distributedly in the networks that shall be analyzed. The proposed system shall be flexible enough to quickly adapt to new requirements like changes in network structure, new data acquisition methods, or the removal or addition of sensors, ideally at runtime. Thus, the integration of these possibly distributed and heterogeneous sensors to an overall system is done via an EDA. Furthermore, the processing of data, like filtering or aggregation of information will to a large extent be performed with means of CEP. Thereby, the sensors correspond to the event producers of the CEP system that is created this way.

In the following we will introduce the details of our proposed approach step by step. We thereby start with outlining some general problems and explaining our proposed solutions to these.

3.1 Architecture

One general problem that needs to be taken into account is the amount and frequency with which data is emitted. Many low-level network analysis methods usually produce very much data with very high frequencies. This, however, can be challenging to handle, especially when considering high-level abstraction infrastructures that possibly span several via WAN connected LANs.

An extreme example of a low-level network analysis approach with respect to the amount and frequency of the generated data is packet capturing. Hence, we purposely choose packet capturing as first example of a data acquisition method because we identified it as potentially most challenging for being applied in the proposed application scenario.

The reason why packet capturing or sniffing is so challenging in this context is, depending on other factors like the sniffing rate (the ratio of captured packets over arrived packets) or bandwidth utilization on the network interface on which packets are captured, that it potentially emits very much data with a very high frequency. As, by definition, all of the data transmitted in packet switched networks, like Ethernet [16] or TCP/IP-based [17,18] networks, is contained in packets, theoretically 100% of the network traffic reaching the interface on which packets are captured can be acquired via packet capturing. Here we neglect very low level overhead added on and below the data link layer, like the Ethernet preamble which typically cannot be recorded with means of packet capturing. Thus, in the extreme case, the bandwidth with which data is emitted equals the bandwidth of the network interface on which packets are captured on.

As also already outlined, many institutional networks and connected to this the respective high-level communication infrastructures span across multiple LANs that are in-turn connected with WANs. Thus, the first, elemental problem when integrating low-level network analysis methods into high-level communication is the potentially huge amount of data that can be emitted by these network

analysis methods. While LANs nowadays typically offer bandwidths in the magnitude of 1 Gbps, site-to-site connections via WANs like Digital Subscriber Line (DSL) usually operate with bandwidths in the magnitude of 10 or 50 Mbps or even below.

Clearly, it is not possible to transfer 1 Gbps of captured data via a 50 Mbps connection. In order to address this issue, we propose an at least 2-tier architecture. Traffic inside the LANs is aggregated and preprocessed in local analysis systems before it is passed to an overarching analysis system. In order to assure a working overall system it is crucial that the information passed to the overarching system is reduced enough such that it can be transferred via slow WAN interconnections. In even bigger setups additional layers can be introduced, effectively building n-tier architectures. This way performance bottle-necks can be efficiently limited while still maintaining the possibility to build big overarching systems. In Fig. 1 a simple example of a 2-tier architecture is shown.

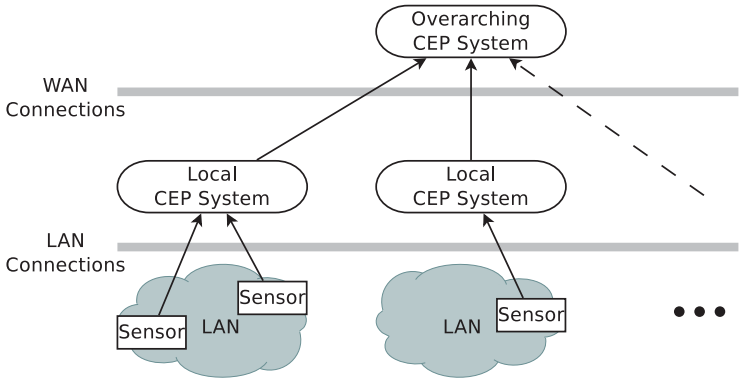


Fig. 1. Simple depiction of a tired architecture that includes multiple LANs via WAN connections.

3.2 Sensors and Data Aggregation

Even in LANs it is not possible to capture 100% of the traffic with maximum bandwidth and transfer the captured data as-is because this would exhaust the resources at hand in the LAN itself. Actually, already sniffing traffic with bandwidths in the magnitude of Gbps is very challenging and either requires special hardware or more sophisticated capturing solutions [19,20]. However, such specialized solutions are a big field of research on their own such that we do not cover this here in-depth. For the research presented here we do not require capture rates near 100% and thus can use common solutions.

A measure to efficiently reduce the amount of data emitted by packet capturing is to output only the packet headers instead of the entire packets and discard the higher protocol payload. With this approach the total amount of data can be

significantly reduced while still maintaining much of the important information. Many data protection regulations prohibit capturing user data anyhow.

Assuming a reassembled (not fragmented) via IP [18] transmitted TCP [17] packet with maximum overall size of 65535 bytes and a maximum accumulated (including TCP, IP, and Ethernet [16] headers), captured (excluding data that is not captured like the Ethernet preamble or footer) header size of 134 bytes the header data only makes up approximately 0.2% of the complete packet, or by just using the header in this example the amount of data can be reduced by a factor of approximately 489. Similar holds for UDP [21] packets as well. In practice packets are usually not sized as big as in this exemplary examination. However, the decrease in size when emitting solely the headers instead of the overall packets is still very significant. Assuming, e. g., much smaller packets with overall size of 2048 bytes and still assuming a maximum accumulated header size of 134 bytes the header makes up roughly 6.54%; correspondingly, the amount of data is still reduced by a factor of about 15.3. Furthermore, in practice headers usually are much smaller than the maximum as used in these exemplary calculations. Note that additional data like a capture header had been neglected here.

Another method for reducing the total amount of data emitted by packet capturing are filter that can be applied to selectively capture only certain packages. Pcap [22] as used in our prototype, e. g., allows very efficient kernel level filtering of packets. This way also the relative sniffing rate, the ratio of captured packets over all arrived packets that should have been captured, can be further increased.

Depending on the actual requirements, these different approaches on reducing the data already during data acquisition can be used combined or selectively. Generally, by reducing the amount of data in the sensors before it is actually fed into the event-driven system the load on the overall system can already be significantly reduced.

Another elemental problem that arises when combining distributed sensor approaches and low-level network analysis is that of a “feedback problem”. By default, the here mentioned data acquisition methods pick up any traffic, including the traffic caused by the communication infrastructure that is required to exchange events. Consequently, the sending of one event results in network traffic, which in turn might be picked up by our sensors, which in turn generate events from this incident which in turn gets send on the network etc. possibly resulting in an infinite loop. Feedback loops are critical as they could result in the network being flooded with self induced traffic. In the worst case such a situation could render the entire network unusable, and thus must be avoided at all costs. In our approach proposed here we brake this loop simply by filtering all traffic on the sensor level that is attributed to our communication infrastructure.

3.3 Communication Infrastructure and Integration

Our proposed system must be easily extendible, allow effortless addition of new data sources, be adaptable to changes in, e. g., network topology, or allow addition and removal of sensors at runtime. In a nutshell, the overall system must be

very flexible and adaptable. Thus a design principle had been chosen that allows very loose coupling between individual components. The general data processing and communication paradigm of our proposed system follows event-driven principles. With this approach event producers (the data sources or sensors), event consumers (the data sinks), as well as most other entities are by and large not concerned with details of the underlying communication infrastructure or addressing individual components. As also mentioned in the previous section we use a multi-tier architecture. Reason is to limit bottlenecks and increase the performance of the compound system.

We chose a message oriented communication infrastructure, namely a message-oriented middleware (MOM) like Java message service (JMS) [23]. MOMs are very well suited as communication infrastructure for EDAs because they also allow loose coupling and enable asynchronous communication. Furthermore, MOMs allow effortless integration for the prototype and can perspectively integrated into other higher-level infrastructures like SOAs. In further research this MOM might be either extended or replaced by an enterprise service bus (ESB) [24]. The choice for a MOM is beneficial as the components are already very loosely coupled and the only parts specific to the current implementation is very little glue-code for attaching the individual components to the MOM.

We exploit some of the features of a MOM to further optimize the performance of our proposed system. Namely, we chose to use a publish-subscriber style notification via so called "topics". We use a distinct topic per event type, in our case, e. g., packets being captured or connections being tracked. All entities that require events being sent to them simply subscribe to the corresponding topics.

This way of "selectively distributing events" is a simple yet efficient way to limit the distribution of events to the components that require the respective events. This results in less traffic in the network, less processing in the message system, and reduced overhead on the event receiving entities in the system. Using, e. g., one general topic for all events would highly increase the network load and messaging overhead as well as the complexity of and load on the event receiving components as these, e. g., would need to actively filter unwanted events.

Furthermore, using a MOM is a very convenient way for establishing n-to-m communication. Neither event sending nor event receiving components need to be aware of the number of receivers or senders; they simply send or receive events to/from the message infrastructure.

By using distinct topics for events also the different components sending and receiving events are highly decoupled. The only thing all entities need to "know" is how to access the overall communication infrastructure. Event sending components then will just create a topic if it doesn't exist yet and will send events to this topic. Components receiving events simply subscribe to the topics of interest. In case no such topic exists yet it will be created as well; however, no events will be available until an event sending component attaches to the corresponding topic and begins sending events.

In further research we will evaluate exploiting the features of an ESB [9,24]. Namely, leveraging the ESB routing functionality to determine even more granularly where events need to be sent. This way we expect that the communication inside the system can be further improved.

The connection to the higher abstraction infrastructures like SOA 2.0 will also be established via the communication infrastructure. Thereby, we currently consider two related solutions. Firstly, what we call "adapters" subscribe to the topics that contain the events intended for further distribution and forward these to the corresponding systems. An "adapter" in this context is a software that subscribes to a topic and forwards these events to the higher-level communication infrastructure. Secondly, the first solution could be extended by providing dedicated topics for forwarding events and than attaching the adapters to these. Via the latter approach an even more granular distribution of events could be realized in a way that events are even more selectively filtered or pre-processed before being sent for further propagation. This detailed filtering and processing is explained in the following section.

3.4 Event Processing

So far the event producers (sensors) as well as the communication infrastructure had been introduced. Finally, the even processing components of our proposed system is where the actual processing of events occurs; i. e., filtering of individual events, transformation of events, or correlation of events takes place. The event processing is especially important to meaningfully pre-process, enrich, transform, and filter events before sending them to higher-level systems.

In our proposed approach the event processing is either done via specialized agents or generally via an attached CEP engine. Depending on the use case either of these two may be beneficial. Also combinations may be used in which data is pre-processed by a special engine, and afterwards is fed into the CEP engine. Generally, arbitrary complex chains of event processing actions can be created. The generic approach on establishing the connection between dedicated processing agents and the CEP engine is to connect these via MOM topics as well.

In section 3.3 on page 7 an approach for selectively distributing events via distinct topics had been introduced. While this can be considered some general way of filtering the selective distribution does not allow more sophisticated fine-grained and detailed filtering of events. In order to filter events, e. g., with respect to more complex criteria like contained data or order of events more sophisticated measures need to be taken. To perform this kind of filtering here we either apply dedicated filtering agents or perform the filtering in the CEP engine.

Another, simpler form of filtering that cannot be done via the aforementioned approach on selective distribution via topics is selectively removing data from events. On the first glance this might be superfluous as event receiving entities can freely select which data to use and which not but there are some use cases that require reducing the data contained within events. At first, reducing data in transmitted events further reduces the network and communication overhead,

and thus may be used for fine-tuning or optimizing the performance. More important, though, excluding data from events might be required due to reasons like data protection regulations or confidentially aspects.

A different form of processing events is to transform events. Essentially, transformation is considered any operation that takes an individual event and creates a new individual event from it. So in one way, the latter approach on filtering event data by removing data from events can be considered a special of transformation. However, usually, transforming events implies changing data formats or data representation; e. g., in our prototype raw header data in form of a byte array is transformed into a map-based structure to ease computation and handling of data.

Finally, CEP can be used to correlate events to each other. In our proposed system this event processing approach is applied in order to detect patterns in the incoming event streams and generate new events based on these. This way events can be very efficiently summarized, and instead of needing to transfer several low-level events to a higher-level system, few more-complex events can be sent instead. This is another key feature for aggregating information and reducing the size of the communicated data, and thus increase the performance.

4 Prototype

For the implementation of our prototype we followed an iterative bottom-up approach. We started with the simplest possible, operational system and then successively added functionality to it. Besides the software engineering aspects of this approach, this choice was also motivated as a test of the flexibility of our approach. By iteratively adding functionality and changing our prototype we could test at first-hand how flexible our proposed approach really is.

4.1 Sensors

For the prototype with which we tested the approach as presented in this paper we exemplarily used the methods of packet capturing and connection tracking [3,25]. In general, we chose these methods because they are inherently event-based; the emitted data already has the nature of events. In case of packet capturing the arrival or capture of a packet is treated as event. In case of connection tracking the change of a connection state is interpreted as event. Furthermore, we chose packet capturing because we identified that especially this method could be challenging to tackle in the context of the proposed approach, e. g., because of the performance aspects associated with it. Connection tracking was chosen because like packet capturing it is an passive, low-level data acquisition method, but one that produces data highly different from the data emitted by packet capturing.

In one of the later iterations, we changed the system by extracting the packet header parsing functionality from the event producer and integrating it into a distinct component. This was on one hand intended as optimization to reduce the

load on the event producer as well as on the network and message infrastructure but also as another "hands-on" test for the flexibility of our proposed approach.

4.2 Communication Infrastructure

In the scope of the system proposed in this paper a message oriented approach on exchanging events had been utilized. A message-oriented middleware (MOM), namely the Apache ActiveMQ JMS system is used in the prototype implementation [23]. Thereby, the code needed for attaching our components to the messaging infrastructure is minimal such that the actual messaging implementation can be effortlessly interchanged at any time. In general, thanks to the choice of an message-driven EDA, our approach also works with other means of communication infrastructures like an ESB.

4.3 Event Processing

As event processing engine we chose the Open Source Esper CEP engine [26]. We evaluated the feasibility of meaningfully processing low-level network analysis data by creating different event patterns. These patterns include, detection of TCP connection setup and tear-down based on packet capturing data, calculating the time-difference between the observation of corresponding ICMP echo request and response packets, and calculating the duration of a TCP connection based on connection tracking data. Additionally, we also implemented a more complex pattern that calculates the duration of a TCP connection. This pattern utilizes two other events that are in-turn themselves derived via CEP. These two patterns detect the TCP connection setup and tear-down based on raw events gathered with packet capturing. Based on these, the duration of a TCP connection is calculated as the difference of the end and start timestamps. Motivation for this was to test more complex patterns, and compare different heterogeneous event sources to each other with respect to CEP.

5 Discussion

We present an approach on integrating low-level network analysis methods into higher-level IT systems like EDA or SOA 2.0. With this approach we present a generic architecture that can be leveraged for integrating arbitrary network analysis methods.

We exemplarily implemented two sensors that employ different low-level network analysis methods. Thereby, we identified some essential problems that arise when integrating low-level network analysis approaches with high-level systems. One issue we discovered was the handling of possibly huge data amounts as emitted by aforementioned network analysis methods. Thereby, especially the limited bandwidth resources in bigger scope network contexts are very problematic. For solving this issue we proposed a scalable multi-tier architecture that can be easily implemented with EDAs and analyzed means for reducing the emitted data

as well as the data that is finally sent to the higher-level systems by reducing the emitted data, filtering, or application of CEP for correlating data into more complex events or a combination of these.

Another problem we identified is the feedback problem that arises when network analysis methods are employed in a distributed context and these analysis methods observe the traffic used for communicating analysis results. Here we proposed a solution by excluding the own network traffic from the network analysis. For both implemented sensors, packet capturing and connection tracking, very efficient ways of filtering data exist. Thereby, the general feedback problem is not limited to an EDA-based approach as sketched in this paper but affects all distributed systems that potentially record their own communication artefacts.

We also evaluated the use of CEP with low-level network analysis data. Therefore, we implemented exemplary patterns and observed their functionality. While most patterns worked as intended we, interestingly, found that the data as generated by the two TCP connection duration patterns differed in some, rare cases. We could not find an explanation for this yet but we will investigate this in further research. Similarly to the performance during the data acquisition, e. g., during sniffing, the performance of the CEP engine may become an issue. During our tests with our prototype we found indications that performance aspects may become problematic when the overall size of the system grows and the number of sensors, and connected to this the overall amount of events that need to be processed, increases. With respect to CEP performance there are already some promising current approaches to increase CEP performance with "in-hardware" CEP systems that use field programmable arrays (FPGAs) [27] or massively parallel computation via general purpose graphical processing units (GPGPUs) [28].

Finally, we tested and evaluated the flexibility of our proposed approach at first-hand by using an iterative implementation approach during the implementation of the prototype. In one of these steps we changed the way data is handled by extracting functionality from one component and putting it in another, newly created component. In the other test we added a new type of sensor that emits highly differing data. In both cases the proposed system architecture proofed to be very flexible and extendible and the required changes were limited to just the directly influenced parts. Other components, the messaging infrastructure, and the CEP engine could be left unchanged. This could be especially nicely seen on the process of externalizing the header parsing which simply required changes in the packet capturing sensor and the addition of the header parsing component. All other components, even those depending on the parsed header events, could be left as-is.

6 Conclusion and Future Work

In this paper we proposed a system for integrating low-level network analysis methods into higher-level IT systems like EDA or SOA 2.0. We identified possible issues and showed ways for solving these.

In order for accessing the practicability of our approach we implemented a prototype that was made up of several components: messaging infrastructure, a CEP engine, a packet capturing sensor, a connection tracking sensor, and a packet header parsing agent. While some of the implementations mostly required integration of pre-existing implementations, others required more implementation effort. With this prototype we could show in first, simple tests that our proposed approach generally works. Repeated tests showed that the system and its components are running stable and produce the expected output. We will use this system as foundation for the next steps in our research.

During our implementation we started with a simple system and extended it step by step. This work was also intended as a test for the flexibility of our proposed approach. It showed that the system is very flexible and extendible. During the different changes we made to the system only the immediately affected components had to be changed.

In future work we plan to further extend the prototype as created during the work presented here. Furthermore, we will perform more in-depth test on the system. While in the work as presented here the emphasis was on the general architecture and concepts as presented in this paper, performance aspects are also a very important topic. Hence, we will also test and evaluate these aspects in more detail in our upcoming future work.

References

1. Shaikh S. A., Chivers H., Nobles P., Clark J. A., Chen H.: Network reconnaissance. Network Security, (2008)
2. Natu, M., Sethi, A.S.: Active Probing Approach for Fault Localization in Computer Networks. 4th IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services, (2006)
3. Sanders, C.: Practical Packet Analysis : Using Wireshark to Solve Real-World Network Problems. No Starch Press, Incorporated, (2007)
4. Webster, S., Lippmann, R., Zissman, M.: Experience Using Active and Passive Mapping for Network Situational Awareness. Fifth IEEE International Symposium on Network Computing and Applications, (2006)
5. Krishnamoorthy, V. and Unni, N.K. and Niranjan, V.: Event-driven service-oriented architecture for an agile and scalable network management system, International Conference on Next Generation Web Services Practices, (2005)
6. Josuttis N. M.: SOA in Practice, O'Reilly Media, Inc., (2007)
7. Nagios Enterprises: Nagios - The Industry Standard in IT Infrastructure Monitoring. Online http://www.nagios.org/ last accessed 03/29/2012
8. Zabbix SIA: Homepage of Zabbix . . . An Enterprise-Class Open Source Distributed Monitoring Solution. Online http://www.zabbix.com/ last accessed 03/29/2012
9. Taylor H., Yochem A., Phillips L., Martinez F.: Event-Driven Architecture: How SOA Enables the Real-Time Enterprise. Addison-Wesley Professional, (2009)
10. Luckham D. C.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley Longman Publishing Co., Inc., (2001)
11. Chandy K. M., Schulte W. R.: Event Processing: Designing IT Systems for Agile Companies McGraw-Hill, (2009)

12. Ficco, M., Romano, L.: A Generic Intrusion Detection and Diagnoser System Based on Complex Event Processing, First International Conference on Data Compression, Communications and Processing (CCP), (2011)
13. Martinez Molina, J.J.,vHernandez Ruiz, M.A., Perez, M.G., Perez, G.M., Gomez Skarmeta, A.F.: Event-Driven Architecture for Intrusion Detection Systems Based on Patterns, Second International Conference on Emerging Security Information, Systems and Technologies SECURWARE '08, (2008)
14. Aniello, L., Di Luna, G. A., Lodi, G., Baldoni, R.: A collaborative event processing system for protection of critical infrastructures from cyber attacks, Proceedings of the 30th international conference on Computer safety, reliability, and security SAFECOMP'11, (2011)
15. Aniello, L., Lodi, G., Baldoni, R.: Inter-domain stealthy port scan detection through complex event processing, Proceedings of the 13th European Workshop on Dependable Computing EWDC '11, (2011)
16. The Institute of Electrical and Electronics Engineers, Inc.: "IEEE Std 802.3-2008 Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications," 2008.
17. Postel J.: Transmission Control Protocol – RFC 793 (Standard). Request for Comments, Internet Engineering Task Force, (1981)
18. Postel J.: Internet Protocol – RFC 791 (Standard). Request for Comments, Internet Engineering Task Force, (1981)
19. Morariu, C. and Stiller, B.: DiCAP: Distributed Packet Capturing architecture for high-speed network links. 33rd IEEE Conference on Local Computer Networks, (2008)
20. Schneider F.: Performance evaluation of packet capturing systems for high-speed networks, Diploma Thesis, Technische Universitat Munchen Fakultat fur Informatik, (2005)
21. Postel J.: User Datagram Protocol – RFC 768 (Standard). Request for Comments, Internet Engineering Task Force, (1980)
22. Tcpdump/Libpcap: TCPDUMP/LIBPCAP public repository, Online http://www.tcpdump.org/ last accessed 03/29/2012
23. Snyder B., Bosanac D., Davies R.: ActiveMQ in Action, Manning Publications (2011)
24. Deng Bo, Ding Kun, Zhang Xiaoyi: A High Performance Enterprise Service Bus Platform for Complex Event Processing. Seventh International Conference on Grid and Cooperative Computing, (2008)
25. Ayuso P.: Netfilter's connection tracking system. LOGIN: The USENIX magazine, (2006)
26. EsperTech Inc.: Esper - Event Stream and Complex Event Processing for Java Reference Documentation Online http://esper.codehaus.org/esper/documentation/documentation.html last accessed 29/04/2012 (2012)
27. Inoue, H., Takenaka, T., Motomura, M.: 20Gbps C-Based Complex Event Processing, International Conference on Field Programmable Logic and Applications (FPL), (2011)
28. Cugola, G., Margara, A.: Low latency complex event processing on parallel hardware, J. Parallel Distrib. Comput., (2012)

Una solución SOA para ejecutar *workflows* científicos en entornos Grid heterogéneos

Sergio Hernández, Javier Fabra, Pedro Álvarez, and Joaquín Ezpeleta

Instituto de Investigación en Ingeniería de Aragón (I3A)
Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza, España
{shernandez,jfabra,alvaper,ezpeleta}@unizar.es

Abstract. La posibilidad de ejecutar un mismo *workflow* científico en distintos entornos Grid heterogéneos todavía es, a día de hoy, un reto abierto. Aunque la orientación a servicios permite allanar el camino, las propuestas existentes exigen un papel activo por parte de los programadores, que deben seleccionar el entorno de ejecución a utilizar en todas las tareas del *workflow* de manera estática. Como consecuencia, estas soluciones limitan la utilización de diversos entornos de computación de forma conjunta. En este trabajo se pretende ir un paso más allá, liberando al programador de la selección del entorno de ejecución y permitiendo ejecutar *workflows* en múltiples entornos de computación heterogéneos. Para ello, se propone un servicio de computación que permite programar *workflows* independientemente del entorno de ejecución y a distintos niveles de abstracción a través de diferentes lenguajes orientados a servicios. Asimismo, el servicio permite integrar varios entornos Grid heterogéneos, mejorando su aprovechamiento mediante una estrategia de *meta-scheduling* basada en simulación que permite decidir el entorno de ejecución más adecuado para cada tarea. Como caso de uso, el *workflow* de análisis *Inspiral* es ejecutado sobre dos entornos heterogéneos, mejorando el rendimiento de la ejecución del *workflow*.

Keywords: *Workflows* científicos, orientación a servicios, *Grid*, integración de sistemas heterogéneos

1 Introducción

El creciente interés de la comunidad científica por automatizar de manera sistemática la ejecución de sus experimentos ha supuesto el impulso definitivo de los *workflows* científicos [1]. Este tipo de *workflow* representa un paradigma para describir, gestionar y compartir análisis científicos abordando la complejidad de los experimentos realizados en este ámbito. Los *workflows* científicos se caracterizan por englobar un elevado número de tareas que se modelan utilizando especificaciones de alto nivel que permiten describir llamadas a servicios Web, *scripts* y tareas locales, tareas de larga duración a ejecutar en entornos de

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 27-40, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

altas prestaciones, *subworkflows* y operaciones de manipulación como servicios de diferentes clases con interfaces de utilización homogéneas. Estas tareas normalmente se estructuran formando grafos acíclicos dirigidos, los cuales incluyen tanto el flujo de datos como el flujo de control del *workflow*, si bien son los datos los que guían generalmente el proceso y determinan cuándo pueden ejecutarse las diferentes tareas que lo componen.

Por otro lado, el paradigma de computación Grid propone el desarrollo de infraestructuras de computación formadas por recursos heterogéneos y geográficamente distribuidos [2]. La capacidad computacional, las comunicaciones en red, la posibilidad de compartir información entre investigadores con intereses comunes y la facilidad de uso de estas infraestructuras gracias a que ofrecen un elevado número de recursos de forma transparente, han promovido su explotación como entornos para la ejecución de *workflows* científicos. No obstante, la programación de *workflows* científicos en este tipo de infraestructuras ha experimentado una fuerte evolución, condicionada por el nivel de abstracción ofrecido por los diferentes sistemas. Los *middleware* que gestionan este tipo de infraestructuras, como gLite [3] o Condor [4], ofrecen su propio lenguaje lo que obliga a que el programador conozcar las características intrínsecas del *middleware* y provoca que los *workflows* estén altamente acoplados al *middleware* y, en muchos casos, a un entorno de ejecución concreto.

La orientación a servicios [5] emergió como una posible solución a estos problemas mediante la aparición de diferentes sistemas de gestión de *workflows* [1]. Ahora, los *middlewares* ofrecen su funcionalidad a través de servicios bajo las premisas y recomendaciones del estándar OGSA [6], y los sistemas de gestión son capaces de ejecutar tareas en los mismos como invocaciones a servicios. Sin embargo, el uso de este tipo de sistemas no logra desacoplar completamente los *workflows* de los *middlewares*. Como consecuencia, mediante esta solución los *workflows* no pueden ser ejecutados en diferentes entornos de computación heterogéneos.

Portales como P-GRADE [7] o HPC-Europa [8] surgen como una solución alternativa. Los portales heredan las virtudes de programar *workflows* con una orientación a servicios y añaden la posibilidad de usar distintos entornos de computación a través de una única interfaz. Sin embargo, estas soluciones están limitadas por el proceso de *scheduling* (esto es, seleccionar en qué entorno se ejecuta cada tarea concreta), que es estático y está guiado por el usuario, lo que implica que el propio usuario debe seleccionar el entorno de ejecución al comienzo del *workflow*, sin que esta información pueda ser modificada durante su ejecución. Esta limitación provoca que no se logre un buen aprovechamiento de los recursos disponibles, ya que el proceso de *scheduling* es muy complejo, y el usuario no suele disponer de información que le ayude en la toma de decisiones. Como consecuencia, las tareas que forman los *workflows* se ven sometidas a elevados tiempos de espera, con la consiguiente reducción del rendimiento obtenido.

Por tanto, las diferentes soluciones existentes hasta el momento o bien obligan a que el *workflow* esté altamente acoplado con un entorno de ejecución concreto, o bien no son capaces de aprovechar la integración de diferentes entornos de

computación. En este trabajo pretendemos resolver los problemas anteriores. Por un lado, se pretende posibilitar la programación de *workflows* de manera independiente del entorno de ejecución y la utilización de diferentes lenguajes ampliamente aceptados por la comunidad científica (Taverna [9], Kepler [10], etc.). Por otro lado, se pretende permitir que diferentes tareas de un mismo *workflow* puedan ejecutarse en diferentes entornos de computación y que el proceso de asignación de tareas a entornos concretos sea llevado a cabo por la infraestructura de integración en base a información dinámica del estado de los recursos. Para conseguir estos objetivos, se ha desarrollado un servicio de computación para la ejecución de *workflows* científicos que encapsula e integra dinámicamente distintos entornos de computación heterogéneos. Además, es capaz de determinar cuál es el entorno más adecuado para la ejecución de cada tarea en base a un mecanismo de *meta-scheduling* basado en simulación, liberando de esta responsabilidad al programador. Finalmente, la interfaz del servicio facilita su uso conjunto con los sistemas de gestión de *workflows* científicos existentes utilizando los lenguajes propuestos por los mismos y el estándar de descripción de trabajos JSDL [11]. Este aspecto facilita la utilización del servicio por diferentes usuarios acostumbrados a diversas herramientas y, en combinación con la posibilidad de delegar la selección del entorno de ejecución al servicio, permite programar *workflows* independientemente del entorno concreto de ejecución.

El resto de este artículo se organiza como sigue. En la Sección 2 se realiza una descripción del servicio indicando su interfaz y de modo de utilización del mismo. En la Sección 3 se muestra la arquitectura interna del servicio y se describen sus componentes principales. La Sección 4 muestra el proceso de *meta-scheduling* utilizado para seleccionar la infraestructura de ejecución en la que ejecutar las tareas enviadas al servicio. La aplicación del servicio a un caso real se detalla en la Sección 5, utilizando como caso de estudio el *workflow* de análisis *Inspiral*. Finalmente, en la Sección 6, se presentan las conclusiones.

2 Descripción del servicio de computación

El servicio de computación ofrecido tiene como objetivo permitir la ejecución de *workflows* científicos en entornos de computación heterogéneos. La Figura 1 refleja las diferentes posibilidades existentes para utilizar el servicio.

El servicio permite trabajar a dos niveles de abstracción: nivel de *workflow* y nivel de tarea. El nivel de abstracción de *workflow* permite solicitar al servicio la ejecución completa de un *workflow*, de forma que el servicio se encarga de la gestión de todo el ciclo de vida del mismo, liberando al usuario de esta labor. Esta alternativa corresponde a la parte izquierda de la Figura 1. Por su parte, trabajar a nivel de abstracción de tarea, permite al programador utilizar sistemas de gestión que controlen el ciclo de vida del *workflow* y solicitar la ejecución de las tareas que componen el mismo bajo demanda. Esta alternativa, reflejada en la parte derecha de la Figura 1, proporciona más flexibilidad al programador y permite integrar en el *workflow* tareas locales y servicios externos.

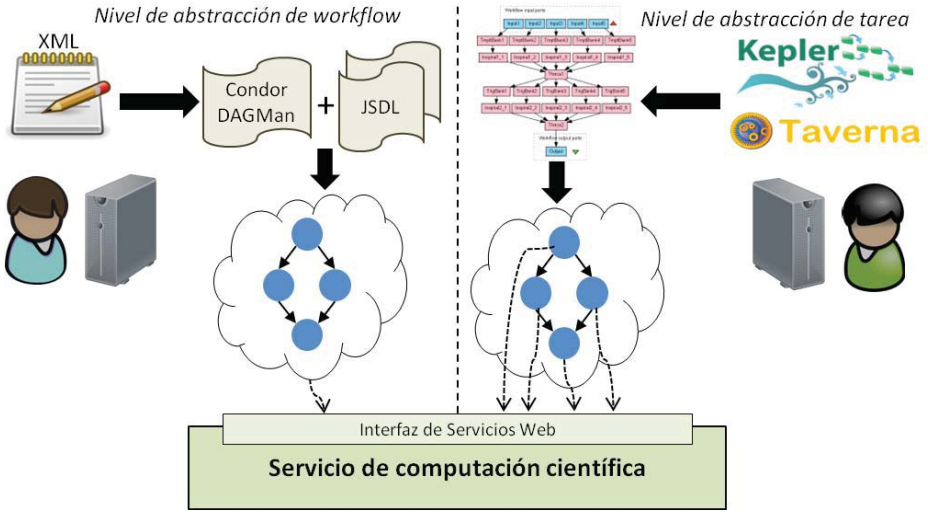


Fig. 1. Interacción de los programadores de *workflows* con el servicio de computación ofrecido a diferentes niveles de abstracción.

En caso de que el programador desee trabajar a nivel de *workflow*, tan sólo es necesaria la descripción de las tareas y las relaciones existentes entre las mismas. Para describir las tareas se utiliza el estándar *Job Submission Description Language* (JSDL), mientras que para describir las relaciones existentes entre las mismas se usa el lenguaje Condor DAGMan. JSDL [11] es un lenguaje estándar propuesto por el *Open Grid Forum*[1] para la descripción textual de tareas, basado en XML y ampliamente utilizado en entornos de computación Grid. La descripción de los trabajos incluye la aplicación a ejecutar, los argumentos pasados a la aplicación, referencias a los datos de entrada y salida involucrados (representados por las URIs correspondientes) y una descripción de los recursos necesarios para ejecutar la tarea (sistema operativo, arquitectura de la máquina, número de CPUs, memoria necesaria, ancho de banda de la red, etc.). Nótese que la descripción de los recursos necesarios no implica que el usuario tenga que especificar los recursos concretos de ejecución, sino que permite indicar las características que deben cumplir dichos recursos para poder ejecutar las tareas. Por su parte, Condor DAGMan [4] es un sistema de gestión de *workflows* que permite especificar de forma textual y sencilla la relación existente entre las tareas de un *workflow*.

Si, por el contrario, el usuario trabaja a nivel de tarea, el servicio puede utilizarse de forma conjunta con un sistema de gestión de *workflows*. La utilización de este tipo de sistemas está muy extendida, en parte debido a que hay sistemas de gestión orientados a una determinada comunidad científica, de forma que presentan facilidades a la hora de construir *workflows* de ese tipo (es

[1] Open Grid Forum, http://www.ogf.org/

el caso de Taverna que está enfocado a la bioinformática). En esta alternativa, el programador especifica las relaciones existentes entre las tareas del *workflow* utilizando el lenguaje proporcionado por el sistema de gestión y envía las tareas al servicio individualmente, siendo el sistema de gestión el encargado de controlar el flujo de ejecución del *workflow*.

En cuanto a la funcionalidad ofrecida, el servicio permite la ejecución de *workflows* programados de forma completamente independiente del entorno de ejecución. Asimismo, los *workflows* pueden programarse a diferentes niveles de abstracción proporcionando una gran flexibilidad al programador. Esta flexibilidad es necesaria por el elevado número de sistemas de gestión utilizados actualmente y la gran diversidad de usuarios existentes, los cuales no sólo están acostumbrados a diferentes herramientas, sino también a diferentes lenguajes y modelos de interacción. Para permitir diferentes tipos de interacción, el servicio incorpora internamente varios componentes de gestión que permiten controlar y gestionar adecuadamente el ciclo de las tareas a ejecutar independientemente del nivel de abstracción utilizado por el usuario (véase la Sección 3). Por otro lado, para poder ejecutar cada tarea en el entorno de computación más adecuado, se incluyen componentes que permiten tomar decisiones basadas en técnicas de simulación (véase la Sección 4).

Para soportar los diferentes modos de utilización, se ofrecen mecanismos de interacción tanto síncronos (bloqueantes) como asíncronos (no bloqueantes), así como operaciones para la monitorización de los *workflows* y la obtención de los resultados junto con el registro de ejecución de los mismos. En el caso de utilizar un modelo de comunicación síncrono, las operaciones ofrecidas por el servicio son las siguientes:

- *execWorkflowS*: Ejecuta de forma completa un *workflow*. Recibe como parámetros la descripción de las tareas y las dependencias existentes entre las mismas y devuelve el registro de la ejecución de las tareas y las referencias a los resultados.
- *execTaskS*: Ejecuta una tarea. Recibe como parámetro la descripción de una tarea y devuelve el registro de ejecución de la tarea y las referencias a los resultados.

En caso de utilizar un modelo de comunicación asíncrono, las operaciones ofrecidas por el servicio son las siguientes:

- *execWorkflowA*: Ejecuta de forma completa un *workflow*. Recibe como parámetros la descripción de las tareas y las dependencias existentes entre las mismas y devuelve como resultado un identificador del *workflow*.
- *execTaskA*: Ejecuta una tarea. Recibe como parámetro la descripción de una tarea y devuelve como resultado un identificador de la tarea.
- *getStatus*: Obtiene el estado de una tarea o *workflow*. Recibe como parámetro el identificador de una tarea o *workflow* y devuelve como resultado el estado de la tarea o *workflow* correspondiente.
- *getResult*: Obtiene el resultado de una tarea o *workflow*. Recibe como parámetro el identificador de una tarea o *workflow* y devuelve como resultado su registro de ejecución y las referencias a los resultados.

Adicionalmente, las operaciones asíncronas permiten indicar una dirección de correo electrónico para avisar al usuario de que la ejecución ha finalizado y facilitar el seguimiento de su estado y la recogida de los resultados.

El servico se ha implementado con tecnologías de Servicios Web, al ser un estándar en la construcción de *middlewares* Grid [6], sencillas de utilizar, y presentar la suficiente flexibilidad como para dar soporte a los diferentes tipos de interacción propuestos. Concretamente, se han desarrollado interfaces SOAP junto con WSDL (para la descripción de las operaciones), y REST. En lo que respecta a los sistemas de gestión soportados, la utilización de un lenguaje estándar de descripción de trabajos permite que el servicio pueda ser utilizado en combinación con cualquier sistema de gestión que permita invocar servicios Web.

3 Arquitectura interna del servicio

La Figura 2 muestra la arquitectura del sistema. En la parte superior, se refleja la interacción con el programador del *workflow*, detallada en la sección anterior. En el centro de la figura se muestra la arquitectura interna del servicio y, finalmente, en la parte inferior se indican las diferentes infraestructuras de computación integradas en el servicio junto con el *middleware* encargado de su gestión. A continuación, analizaremos en profundidad la arquitectura interna del servicio y describiremos las infraestructuras de computación utilizadas.

Internamente, el servicio se ha diseñado de forma que las diferentes componentes se encuentran desacopladas y pueden ser sustituidas, adaptadas o modificadas de forma transparente para el usuario. Concretamente, el servicio está formado por un *broker* de recursos y un conjunto de componentes de gestión. El *broker* constituye el núcleo del servicio, encargándose de gestionar la interacción con el exterior, conocer el estado de las infraestructuras y permitir la comunicación entre los diferentes componentes del sistema. Por su parte, las componentes de gestión ofrecen diferentes funcionalidades encaminadas a la gestión de las tareas de los *workflows*.

El *broker* está formado por un repositorio de mensajes y una infraestructura de mediadores. La comunicación entre las diferentes componentes se realiza a través de mensajes que contienen información de diversa naturaleza y que se almacenan en el repositorio. La implementación del repositorio de mensajes se ha inspirado en el modelo de coordinación Linda [12]. Los mensajes se codifican como tuplas y son almacenados en un espacio de tuplas. La interfaz del repositorio proporciona una operación de escritura de tuplas y dos operaciones bloqueantes de lectura (destructiva y no destructiva) de acuerdo a la semántica de Linda. Este enfoque se traduce en que cada componente sólo interacciona con el repositorio y no tiene que comunicarse directamente con otras componentes. Por tanto, el uso del repositorio de mensajes permite desacoplar las diferentes componentes de la capa de ejecución. Como resultado, se otorga flexibilidad al servicio al permitir que se añadan, modifiquen o eliminen componentes de forma dinámica sin que esto afecte al resto de componentes.

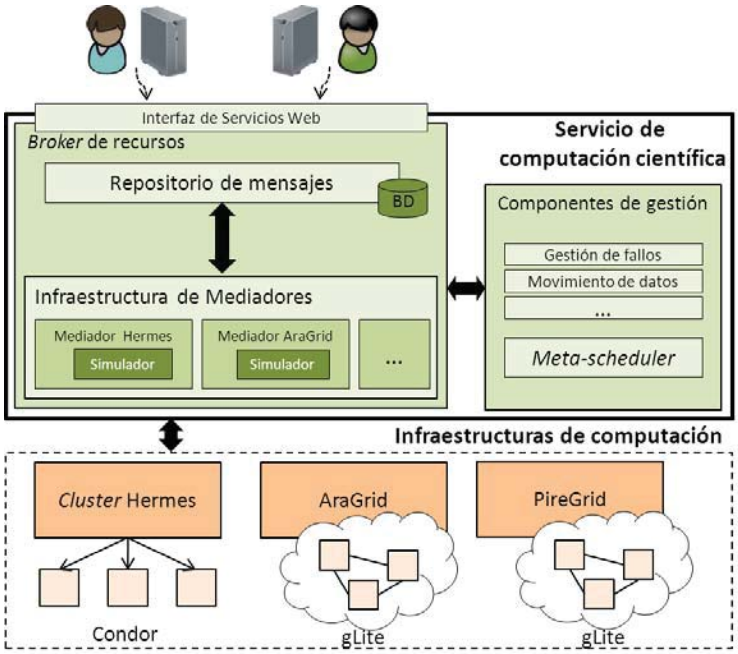


Fig. 2. Arquitectura del servicio propuesto e interacción del mismo con el exterior.

Por su parte, los mediadores encapsulan la heterogeneidad de los *middlewares*, teniendo completo conocimiento de sus capacidades y características. Este diseño elimina la necesidad de que el *broker* tenga que estar estrechamente acoplado con la tecnología concreta del entorno Grid, permitiendo incorporar diferentes entornos de computación heterogéneos de forma sencilla y transparente para el programador de *workflows*. Internamente, el mediador es responsable de: i) tener información completa del entorno Grid que encapsula; ii) interaccionar con el repositorio de tuplas para obtener tareas a ejecutar; iii) enviar tareas al *middleware* para su ejecución y controlar la transferencia de los datos de entrada y de salida; y iv) insertar tuplas en el repositorio de mensajes con el resultado de la ejecución de las mismas para que sea tratada por la componente adecuada. Se ha implementado un mediador para cada uno de los *middlewares* (Condor y gLite) utilizados por los entornos de computación disponibles, los cuales son, además, dos de los *middlewares* más utilizados en entornos Grid.

En cuanto a las componentes de gestión, éstas ofrecen diferentes funcionalidades encaminadas a gestionar el ciclo de vida de los *workflows* ejecutados. Se ha desarrollado una componente de gestión de fallos y una componente de movimiento de datos. El procedimiento de integración de estas componentes es similar al utilizado en los mediadores. Cada componente de gestión interacciona con el repositorio de mensajes para retirar mensajes con la etiqueta asociada a esa componente y procesarlos. Por tanto, la utilización de estas componentes

puede ser debida a la necesidad de un procesado concreto (p. ej. *meta-scheduling*) o como respuesta al resultado de otra componente (p. ej. gestión de fallos), permitiendo la composición dinámica de complejas cadenas de acción. Con la integración de estas componentes se consigue gestionar de forma completa el ciclo de vida de un *worfklow*. En [13] se pueden consultar más detalles sobre la implementación de los mediadores y las componentes de gestión.

A modo de ejemplo, para que el lector comprenda la interacción existente entre las componentes del servicio, mostraremos el proceso seguido al utilizar *execTaskA* para ejecutar una tarea de forma asíncrona. En primer lugar, la descripción de la tarea se almacena en el repositorio de mensajes. A continuación, el *meta-scheduler* obtiene la tarea y solicita a los mediadores que simulen la ejecución de la misma. Con los resultados de las simulaciones, el *meta-scheduler* decide cuál es la infraestructura más adecuada para su ejecución. Dicha información es almacenada en el repositorio y recuperada por el mediador correspondiente. Antes de ejecutar la tarea, el mediador solicita el movimiento de los datos de entrada necesarios. Una vez transferidos, el mediador envía la tarea al Grid que representa para que se ejecute. Cuando la tarea finaliza o falla, el mediador solicita el movimiento de los datos de salida a su ubicación final, recupera el registro de ejecución de la tarea e introduce dicha información en el repositorio de mensajes. Si la tarea ha finalizado correctamente, se envía un correo electrónico al usuario (sólo si se indicó al enviar la tarea) y la información queda almacenada en el repositorio de mensajes hasta que sea obtenida por el usuario a través de la operación *getResult*. Si por contra, la tarea ha fallado, la componente de gestión de fallos obtiene la causa del fallo y toma alguna decisión al respecto, por ejemplo, volver a ejecutar la tarea en otra infraestructura o notificar al usuario del error que se ha producido. En caso de que la tarea sea reejecutada, se repite el proceso, mientras que si se decide avisar al usuario del fallo, se actúa de la misma manera que en el caso de que la tarea finalice correctamente.

En lo que corresponde a las infraestructuras de computación, se han integrado: el *cluster* Hermes del Instituto de Investigación en Ingeniería de Aragón (I3A)[2], gestionado utilizando el *middleware* Condor, y dos Grids pertenecientes a la Iniciativa Grid Europea (EGI)[3]: AraGrid[4] y PireGrid[5], gestionados por el *middleware* gLite y administrados por el Instituto de Biocomputación y Física de Sistemas Complejos (BIFI)[6]. A pesar de su heterogeneidad, gracias a la infraestructura de mediadores y al diseño desacoplado de las componentes del sistema su integración resulta sencilla. Asimismo, la utilización de un *meta-scheduler* y otras componentes de gestión permite utilizar las infraestructuras de forma conjunta, siendo este proceso transparente para el usuario. Como resul-

[2] Instituto de Investigación en Ingeniería de Aragón (I3A), http://i3a.unizar.es/
[3] EGI: The European Grid Initiative, http://www.egi.eu/
[4] AraGrid, http://www.araGrid.es/
[5] PireGrid, http://www.pireGrid.eu/
[6] Instituto de Biocomputación y Física de Sistemas Complejos (BIFI), http://bifi.es/es/

tado, se consigue dotar al servicio de una elevada potencia de cálculo y mejorar el rendimiento de los *workflows* ejecutados.

4 *Meta-scheduling* basado en simulación

La introducción de una componente de *meta-scheduling* proporciona nuevas oportunidades respecto a estrategias de planificación básicas que obligan al usuario a indicar la plataforma de ejecución, seleccionan una infraestructura de forma aleatoria o elijen una infraestructura en base a criterios estáticos. Se han propuesto multitud de estrategias en la literatura [14]. En general, estas estrategias buscan optimizar algún tipo de función objetivo como, por ejemplo, el coste de utilización de los recursos o el tiempo de ejecución. En nuestro caso concreto, el algoritmo utilizado por la componente de *meta-scheduling* tiene como objetivo minimizar el tiempo de ejecución, utilizando para ello los resultados proporcionados por los simuladores. En cualquier caso, la discusión de la mejor estrategia de *meta-scheduling* posible queda fuera del alcance de este trabajo.

La Figura 3 muestra el proceso que supone la ejecución de las tareas de un *workflow* utilizando un *meta-scheduler*. Inicialmente, las tareas pendientes (abstractas), almacenadas en el repositorio de mensajes, son recuperadas por el *meta-scheduler*. A continuación, esta componente determina las infraestructuras capaces de ejecutar dichas tareas y solicita a los mediadores correspondientes que simulen su ejecución. Los mediadores realizan la simulación, utilizando *workloads* construidos en tiempo de ejecución, y devuelven el resultado al *meta-scheduler*. Cuando el *meta-scheduler* ha recibido el resultado de todas las simulaciones, elige la infraestructura más adecuada en base al algoritmo de optimización utilizado y almacena dicha información en la descripción de la tarea, la cual pasa a ser una tarea concreta. Finalmente, el *meta-scheduler* envía la tarea al repositorio de mensajes para que sea recuperada por el mediador correspondiente y ejecutada.

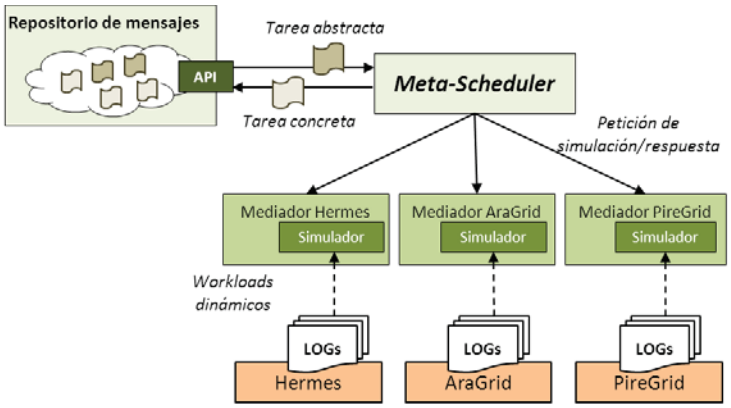


Fig. 3. Componente arquitectural para realizar *scheduling* basado en simulación.

Para soportar este proceso, los mediadores se han extendido mediante la integración de un simulador en cada mediador. El simulador es capaz de: i) modelar el entorno de computación (recursos computacionales, memoria, ancho de banda de la red, usuario, política de *scheduling*, etc.); ii) seleccionar el *workload* más adecuado para representar el comportamiento real de cada entorno de ejecución y el estado actual de sus recursos (para crear estos *workloads* se han utilizado registros de ejecución reales); y, finalmente, iii) simular la ejecución de tareas midiendo parámetros como el tiempo de ejecución, el tiempo de transferencia de los datos, el tiempo de encolamiento, la memoria consumida, etc. La integración del simulador como componente interno de cada mediador permite que los mismos sean capaces de manejar diferentes situaciones y personalizar la simulación dependiendo del estado concreto de la infraestructura. En cualquier caso, el simulador es accedido a través de una API bien definida, de forma que añadir nuevos simuladores es sencillo y sólo implica modificar el modelo de la infraestructura y la política de *scheduling* utilizada por la misma .

Como base para los simuladores desarrollados se ha utilizado Alea [15]. Alea es un simulador basado en eventos y construido sobre GridSim [16]. Alea extiende GridSim proporcionando un *scheduler* centralizado, mejorando algunas funcionalidades y aumentando la escalabilidad y la velocidad de la simulación. Además, Alea proporciona un entorno de experimentación fácil de configurar y utilizar, el cual ayuda en la adaptación del simulador a las nuevas infraestructura añadidas al servicio. La implementación de Alea ha sido extendida para soportar tanto la política de scheduling basada en prioridades utilizada por *Condor* como la política jerárquica utilizada por *gLite*.

Un aspecto clave del proceso es la creación del *workload* que indica las tareas a simular. La importancia de utilizar un *workload* apropiado ha sido identificada en varios trabajos [17, 18]. El uso de un *workload* erróneo puede provocar que los resultados de la simulación no se ajusten al comportamiento real y, por tanto, lleven a tomar malas decisiones de *meta-scheduling*. En nuestro caso, los *workloads* se crean en tiempo de ejecución mediante la agregación de las tareas que se quieren simular, las tareas que se están ejecutando actualmente en la infraestructura y una serie de tareas que representan la carga esperada de la infraestructura.

El modelo del Grid también se proporciona como entrada a la simulación. Este modelo incluye los recursos *hardware* que incluye el entorno de ejecución y las características de la red. El modelo se adapta automáticamente en el momento en el que se va a realizar una simulación, indicando los recursos que se encuentran caídos para reflejar el estado actual del entorno. Finalmente, el simulador está compuesto por diferentes componentes que permiten abordar la complejidad de los entornos de ejecución y entre los que destaca el *Scheduler*, la componente que selecciona el recurso concreto en el que ejecutar cada tarea del *workload*. En [19] se puede consultar el proceso de simulación en detalle.

5 Caso de estudio: Inspiral

En esta sección desplegaremos y ejecutaremos el *workflow* científico de análisis *Inspiral* con la solución propuesta. El *workflow* de análisis Inspiral es un *workflow* científico que analiza e intenta detectar ondas gravitacionales producidas por varios eventos en el universo utilizando datos obtenidos de la coalescencia de sistemas binarios compactos como estrellas binarias de neutrones y agujeros negros [20]. La Figura 4 muestra la estructura del *workflow* (Figura 4-a) y su implementación en Taverna (Figura 4-b). Como puede observarse, la relación entre cada una de las tareas que forman una fase en el diseño de alto nivel y la implementación correspondiente del *workflow* en Taverna es inmediata, siendo muy sencilla la composición del experimento. Internamente, cada una de las cajas que representan las tareas en Taverna encapsula varias operaciones sencillas como son la generación de la descripción de las tareas y las invocaciones al servicio.

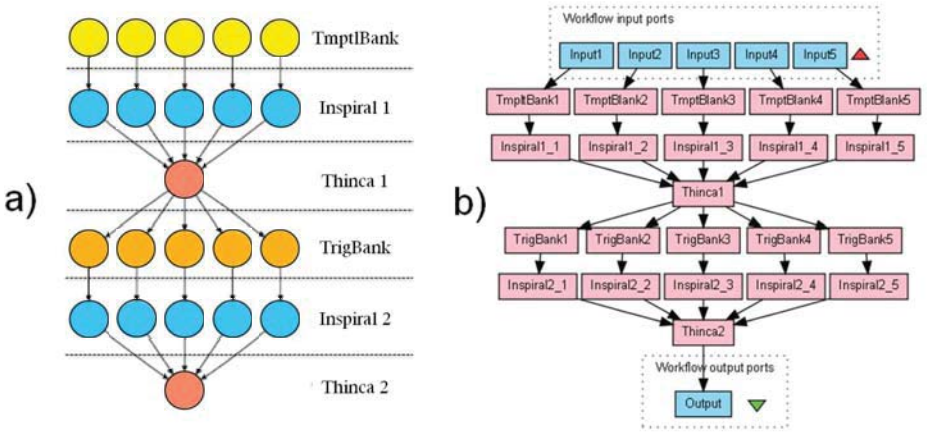


Fig. 4. *Workflow* científico de análisis *Inspiral*. a) Descripción de alto nivel. b) Implementación en Taverna.

El experimento consta de diferentes fases, cuya descripción detallada puede consultarse en [19, 20], que realizan el procesamiento de grandes conjuntos de mediciones generadas por un conjunto de sensores y detectores. Las tareas que invocan al servicio *Inspiral* son las más complejas en términos computacionales y las que más recursos demandan.

El *workflow Inspiral* se puede ejecutar tanto en Hermes como en AraGrid. Sin embargo, ambas infraestructuras muestran una tendencia a tener diferentes niveles de carga a lo largo del día, lo que provoca que sea más viable enviar los trabajos a una infraestructura en ciertos momentos y viceversa. Los detalles de los diferentes niveles de carga pueden consultarse en [19]. Evidentemente, éste no es el único criterio a considerar ya que el rendimiento de una plataforma

Grid depende de muchos factores y su análisis es complejo. La utilización de un simulador como herramienta de decisión permite lidiar con esta complejidad y mejorar el rendimiento obtenido en la ejecución del *workflow* como se muestra en la Figura 5. La figura muestra el tiempo de ejecución total de cada etapa para el *workflow Inspiral* ejecutado de forma completa en cada infraestructura (la barra izquierda corresponde a Hermes mientras que la barra derecha corresponde a AraGrid) y ejecutado utilizando el servicio y la política de *meta-scheduling* descrita anteriormente (barra central de la figura). Los resultados muestran que, para el caso del experimento Inspiral, la utilización del servicio permitiría obtener una mejora del 59% respecto a la ejecución del *workflow* en Hermes y un 111% respecto a la ejecución del mismo en AraGrid.

Respecto a la sobrecarga que introduce la simulación en términos de tiempo de ejecución, el proceso de simulación de Hermes es más complejo y tarda entre 3 y 4 minutos para una bolsa de 10000 tareas, mientras que para AraGrid lleva en torno a 1 minuto. Por otro lado, las transferencias de datos entre infraestructuras usan enlaces de alta velocidad lo que implica que el tiempo de ejecución disminuya a pesar de la sobrecarga introducida.

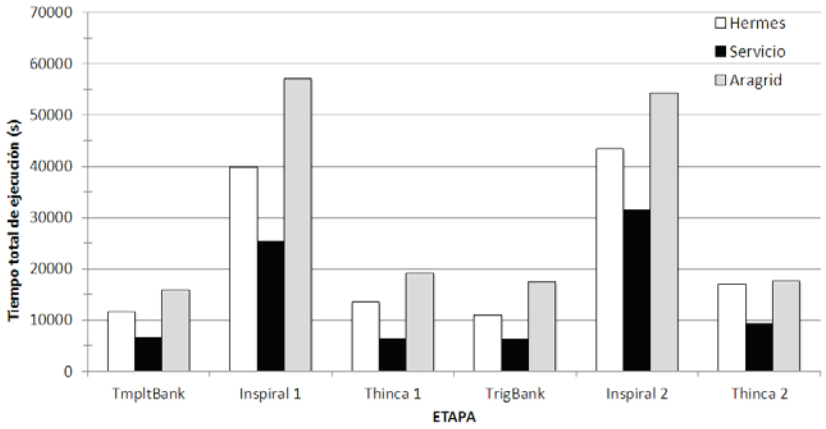


Fig. 5. Resultados experimentales para el *workflow* Inspiral.

6 Conclusiones

En este artículo se ha propuesto un servicio de computación para la ejecución de *workflows* científicos que resuelve varios de los problemas abiertos en el contexto de la ejecución de *workflows* científicos en infraestructuras Grid en lo que atañe al modelado de los *workflows* y a la ejecución de los mismos. En primer lugar, el servicio permite ejecutar *workflows* programados de manera independiente del entorno de ejecución. Esto permite liberar al programador de los detalles de bajo nivel referentes a la infraestructura y la interacción con el

middleware, lo que facilita la creación y reutilización del *workflow*. Asimismo, la posibilidad de utilizar el servicio junto con un sistema de gestión de *workflows* permite ejecutar *workflows* programados en diferentes lenguajes, facilitando el uso del servicio por usuarios con diferentes conocimientos y necesidades. En segundo lugar, el diseño flexible y desacoplado propuesto permite integrar diversas infraestructuras de computación heterogéneas de forma dinámica y utilizar las mismas conjuntamente, dotando al servicio de una elevada potencia computacional. La inclusión de un *meta-scheduler* y técnicas de simulación permite decidir de forma dinámica la infraestructura más adecuada para ejecutar cada tarea, permitiendo sacar un mejor partido a las infraestructuras disponibles y obteniendo una mejora en el rendimiento de los *workflows* ejecutados, como puede observarse en la aplicación del mismo al *workflow* de análisis *Inspiral*.

Actualmente se está trabajando en la mejora de diferentes aspectos del servicio. Por un lado, se pretenden integrar más infraestructuras de computación. En particular se plantea la utilización de entornos de computación Cloud como Amazon EC2. Por otra parte, se pretende aumentar la precisión de las simulaciones, para poder tomar mejores decisiones y aumentar el rendimiento de los *workflows* ejecutados. En esa misma línea, se pretende estudiar el rendimiento que ofrecen diferentes algoritmos de *meta-scheduling*. Finalmente, se pretende aplicar el servicio a la resolución de problemas complejos desde un punto de vista computacional, como el análisis del comportamiento de *workflows* científicos anotados semánticamente.

Agradecimientos

Este trabajo se ha financiado parcialmente gracias al proyecto de investigación del Ministerio de Educación y Ciencia TIN2010-17095 y al Fondo Social Europeo (DGA-FSE) del Fondo Europeo de Desarrollo Regional (FEDER).

Referencias

1. Yu, J., Buyya, R.: A taxonomy of workflow management systems for Grid computing. J. Grid Comput. **3** (2005) 171–200
2. Foster, I., Kesselman, C.: The Grid 2: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2003)
3. Laure, E., Gr, C., Fisher, S., Frohner, A., Kunszt, P., Krenek, A., Mulmo, O., Pacini, F., Prelz, F., White, J., Barroso, M., Buncic, P., Byrom, R., Cornwall, L., Craig, M., Meglio, A.D., Djaoui, A., Giacomini, F., Hahkala, J., Hemmer, F., Hicks, S., Edlund, A., Maraschini, A., Middleton, R., Sgaravatto, M., Steenbakkers, M., Walk, J., Wilson, A.: Programming the Grid with gLite. In: Computational Methods in Science and Technology. (2006)
4. Thain, D., Tannenbaum, T., Livny, M.: Distributed computing in practice: the Condor experience: Research articles. Concurrency and Computation: Practice and Experience **17** (2005) 323–356
5. Foster, I.: Service-Oriented Science. Science **308** (2005) 814–817

6. Gannon, D., Plale, B., Christie, M., Fang, L., Huang, Y., Jensen, S., Kandaswamy, G., Marru, S., Pallickara, S.L., Shirasuna, S., Simmhan, Y., Slominski, A., Sun, Y.: Service oriented architectures for science gateways on Grid systems. In: ICSOC. (2005) 21–32
7. Farkas, Z., Kacsuk, P.: P-GRADE Portal: A generic workflow system to support user communities. Future Generation Comp. Syst. **27** (2011) 454–465
8. Oleksiak, A., Tullo, A., Graham, P., Kuczynski, T., Nabrzyski, J., Szejnfeld, D., Sloan, T.: HPC-Europa: Towards uniform access to European HPC infrastructures. In: 6th IEEE/ACM International Conference on Grid Computing (GRID 2005), IEEE (2005) 308–311
9. Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M.R., Li, P., Oinn, T.: Taverna: a tool for building and running workflows of services. Nucleic acids research **34** (2006) W729–732
10. Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E.A., Tao, J., Zhao, Y.: Scientific workflow management and the Kepler system. Concurrency and Computation: Practice and Experience **18** (2006) 1039–1065
11. Anjomshoaa, A., Brisard, F., Drescher, M., Fellows, D., Ly, A., Mcgough, S., Pulsipher, D., Savva, A.: Job Submission Description Language (JSDL) Specification, Version 1.0. Technical report, Global Grid Forum (2005)
12. Carriero, N., Gelernter, D.: Linda in context. Commun. ACM **32** (1989) 444–458
13. Fabra, J., Hernández, S., Álvarez, P., Ezpeleta, J.: A Framework for the Flexible Deployment of Scientific Workflows in Grid Environments. In: Third International Conference on Cloud Computing, GRIDs, and Virtualizations – CLOUD COMPUTING 2012. (2012)
14. Li, Y., Lan, Z.: A survey of load balancing in Grid computing. In: Proceedings of the First international conference on Computational and Information Science. CIS'04, Berlin, Heidelberg, Springer-Verlag (2004) 280–285
15. Dalibor Klusácek, H.R.: Alea 2 – job scheduling simulator. In: Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools 2010), ICST (2010)
16. Sulistio, A., Cibej, U., Venugopal, S., Robic, B., Buyya, R.: A toolkit for modelling and simulating data Grids: an extension to GridSim. Concurrency and Computation: Practice and Experience **20** (2008) 1591–1609
17. Feitelson, D.: Workload modeling for performance evaluation. In: Performance Evaluation of Complex Systems: Techniques and Tools. Springer, Berlin / Heidelberg (2002) 114–141
18. Li, H., Groep, D., Wolters, L.: Workload characteristics of a multi-cluster supercomputer, Springer Verlag (2004) 176–193
19. Hernández, S., Fabra, J., Álvarez, P., Ezpeleta, J.: A Simulation-based Scheduling Strategy for Scientific Workflows. In: Second International Conference on Simulation and Modeling Methodologies, Technologies and Applications – SIMULTECH 2012. (2012)
20. Taylor, I.J., Deelman, E., Gannon, D.B., Shields, M.: Workflows for e-Science: Scientific Workflows for Grids. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)

CumuloNimbo: Una Plataforma como Servicio con Procesamiento Transaccional Altamente Escalable*

Ricardo Jiménez-Peris, Marta Patiño-Martínez, Ivan Brondino
{rjimenez,mpatino,ibrondino}@fi.upm.es

Facultad de Informática
Universidad Politécnica de Madrid

Resumen El modelo de computación en la nube (*cloud computing*) ha ganado mucha popularidad en los últimos años, prueba de ello es la cantidad de productos que distintas empresas han lanzado para ofrecer software, capacidad de procesamiento y servicios en la nube. Para una empresa el mover sus aplicaciones a la nube, con el fin de garantizar disponibilidad y escalabilidad de las mismas y un ahorro de costes, no es una tarea fácil. El principal problema es que las aplicaciones tienen que ser rediseñadas porque las plataformas de computación en la nube presentan restricciones que no tienen los entornos tradicionales. En este artículo presentamos *CumuloNimbo*, una plataforma para computación en la nube que permite la ejecución y migración de manera transparente de aplicaciones multi-capa en la nube. Una de las principales características de CumuloNimbo es la gestión de transacciones altamente escalable y coherente. El artículo describe la arquitectura del sistema, así como una evaluación de la escalabilidad del mismo.

1. Introducción

Hoy en día muchas aplicaciones se programan con una arquitectura multicapa, generalmente de tres capas, en la que se separa por un lado la capa de presentación, la capa de aplicación y por otro, la capa de almacenamiento de datos. Esta separación implica el uso de distintos sistemas para la programación de cada capa (servidor web, servidor de aplicaciones y base de datos) que se ejecutan en un entorno distribuido y que, a su vez, pueden estar cada uno de ellos replicado para tolerar fallos y escalar a medida que aumenta el número de clientes. En la actualidad muchas aplicaciones se ofrecen en la *nube* ejecutando remotamente, como por ejemplo el correo electrónico o los documentos de gmail. El único requisito para acceder a estos servicios es disponer de una conexión a internet. A este modelo se le denomina *software como servicio* (SaaS). También se han desarrollado *plataformas como servicio* (PaaS) como por ejemplo windows Azure, en las que las aplicaciones se ejecutan remotamente haciendo uso de los servidores, redes y almacenamiento del proveedor de la plataforma.

Este artículo presenta CumuloNimbo[1] [7], una plataforma como servicio (PaaS), para aplicaciones multi-capa que prevé un procesamiento transaccional ultra-escalable proporcionando el mismo nivel de consistencia y transparencia que un sistema de base de datos relacional tradicional. La mayoría de los sistemas actuales recurren al *sharding* para obtener escalabilidad en el procesamiento transaccional. *Sharding* es una técnica en la cual la base de datos se divide en varios fragmentos (particiones) que funcionan como bases de datos independientes compartiendo el esquema de la base de datos original. *Sharding* es técnicamente sencillo pero no es ni sintáctica ni semánticamente transparente. La transparencia sintáctica se pierde porque las aplicaciones tienen que ser reescritas con transacciones a las que sólo se les permite acceder a una de las particiones. La transparencia semántica se pierde, porque las propiedades ACID previamente proporcionadas

* This work was partially funded by the Spanish Research Council (MiCCIN) under CloudStorm TIN2010-19077, by the Madrid Research Foundation, Clouds project S2009/TIC-1692 (cofunded by FEDER & FSE), and CumuloNimbo project FP7-257993.

[1] http://cumulonimbo.eu/

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 41-47, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

por transacciones sobre conjuntos de datos arbitrarios se pierden. Recientemente se han propuesto alternativas al *sharding* [3],[11], pero son soluciones para estructuras de datos especializadas [3] o no han sido diseñadas para sistemas online que requieren tiempos de respuesta rápidos [11].

El objetivo de la plataforma CumuloNimbo es dar soporte a aplicaciones con procesamiento transaccional online ofreciendo las mismas garantías que las ofrecidas por las bases de datos relacionales tradicionales, es decir propiedades ACID estándar, y al mismo tiempo garantizando una escalabilidad como la de las plataformas que emplean *sharding*. CumuloNimbo proporciona el mismo entorno de programación que un servidor de aplicaciones Java EE, consiguiendo una transparencia sintáctica y semántica completa para las aplicaciones.

2. Trabajo Relacionado

Recientemente se han publicado varios trabajos que proponen protocolos para el procesamiento transaccional en entorno cloud.

En CloudTPS [12] los datos son particionados (*sharding*) para poder escalar el procesamiento de transacciones. Las claves primarias a las que accede una transacción tienen que ser suministradas al inicio de la misma. Microsoft SQL Azure [10] sigue un modelo similar en el que los datos tienen que ser particionados manualmente (*sharding*) y las transacciones no pueden acceder a más de una transacción. ElasTraS [5] es una base de datos elastic en la que las transacciones solo pueden acceder a una partición de los datos.

Google Megastore [1] proporciona transacciones serializables sobre Bigtable. Los programadores tienen que crear grupos de datos y una transacción solo puede acceder a un grupo.

Los principales problemas a la hora de construir una base de datos sobre S3 (el servicio de almacenamiento de Amazon) son presentados en [4]. Los autores argumentan que la propiedad de aislamiento de las transacciones no se puede suministrar ya que el contador necesario para implementar aislamiento *snapshot* se convierte en un cuello de botella y punto único de fallo. El diseño de CumuloNimbo evita estos dos problemas.

Deuteronomy [9] desacopla el almacenamiento del procesamiento transaccional. Las transacciones implementan todas las propiedades ACID y pueden acceder a cualquier conjunto de datos. Adopta un enfoque centralizado para el procesamiento de transacciones que puede convertirse en cuello de botella. En este caso, se pueden desplegar varios gestores de transacciones, pero cada gestor solo puede modifica datos disjuntos.

Percolator [11] es la propuesta de Google para el procesamiento transaccional de grandes cantidades de datos. Al igual que CumuloNimbo proporciona aislamiento*snapshot*. Aunque no tiene interfaz de base de datos relacional y está diseñada para realizar procesamiento batch donde el tiempo de respuesta no es la métrica principal.

CumuloNimbo proporciona la funcionalidad de una base de datos relacional, es escalable, transparente para los usuarios (no hay particionamiento) y tolerante a fallos (implementado en parte por el almacenamiento).

3. Arquitectura de CumuloNimbo

La arquitectura de CumuloNimbo consta de múltiples capas (figura 1), cada una de las cuales tiene una funcionalidad específica. Para conseguir el rendimiento deseado, cada capa puede ser escalada añadiendo más servidores. Para obtener la calidad de servicio necesaria en términos de tiempos de respuesta máximos, las capas más altas deberán procesar tantas peticiones como sea posible sin propagarlas al resto de las capas. Los principales retos son mantener las propiedades transaccionales en todas las capas y conseguir que el procesamiento de transacciones (control de concurrencia y persistencia) sean escalables.

Las instancias del servidor de aplicaciones constituyen la primera capa (*Application Server*). Actualmente, el sistema emplea Java EE, pero cualquier otra tecnología de servidor de aplicaciones podría ser integrada, como por ejemplo .NET. En nuestro prototipo actual usamos el servidor de

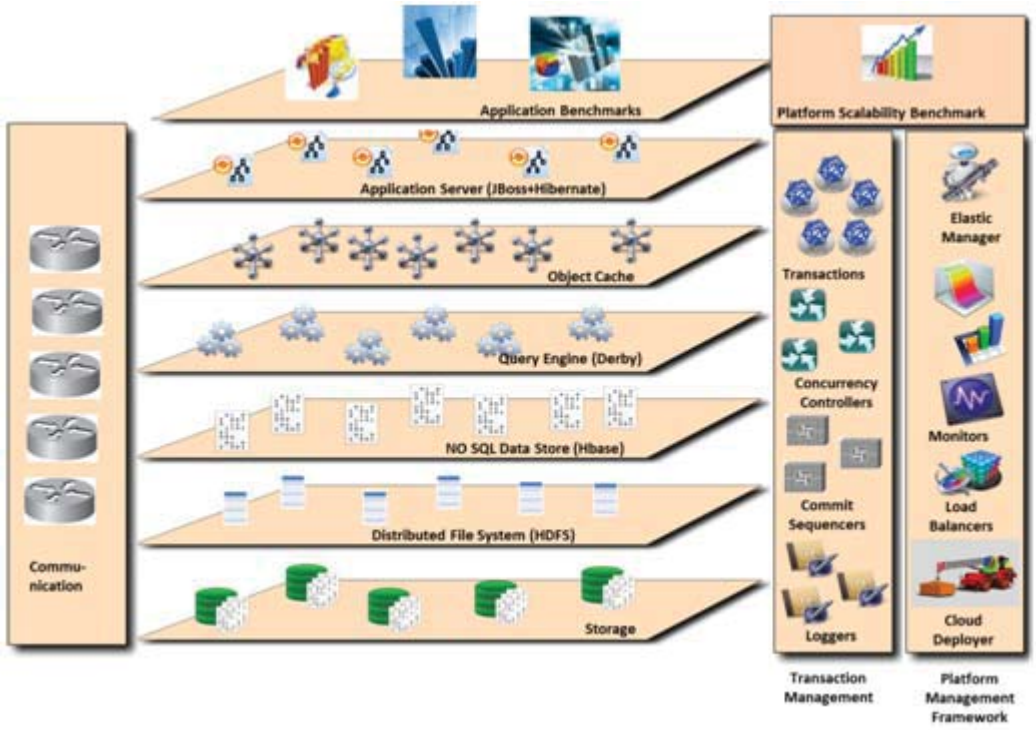


Figura 1. Arquitectura CumuloNimbo

aplicaciones JBoss[2] junto con Hibernate[3] para hacer la transformación objetos a relacional y a la inversa. El procesamiento de transacciones local de JBoss ha sido inactivado y sustituido por el gestor de transacciones de CumuloNimbo. El número de servidores de aplicaciones dependerá del consumo de CPU de la aplicación y de la cantidad de peticiones concurrentes que un único servidor de aplicaciones puede manejar.

La segunda capa está constituida por una caché distribuida de objetos (*Object Cache*). La caché es compartida por todas las instancias del servidor de aplicaciones. La caché puede alojarse en tantos nodos como sean necesarios para mantener los objetos en memoria principal, incluso la base de datos completa, si fuera necesario. Cuando se solicite el acceso a un objeto por la aplicación, Hibernate buscará primero el objeto en la caché distribuida, y sólo si no está disponible en la misma, lo solicitará a la capa de base de datos. La caché también soporta el almacenamiento de los resultados de consultas.

La tercera capa es la capa del motor de consultas SQL (*Query engine*). Esta capa es accedida si: (a) un objeto es accedido, pero el objeto no está almacenado en la caché de objetos, (b) se envía una consulta SQL que no puede ser contestada por la caché, (c) los cambios en los objetos tienen que ser almacenados cuando compromete una transacción para hacerlos persistentes. El número de motores de consultas dependerá del número de consultas complejas que son ejecutadas concurrentemente y de la cantidad de peticiones concurrentes que un motor de consultas pueda manejar. El motor de consultas empleado es el proporcionado por el sistema de gestión de bases de datos Derby[4]. Éste contiene el planificador de consultas y los componentes de ejecución y optimización de las mismas, pero toda la gestión transaccional ha sido desactivada.

La cuarta capa está constituida por el almacenamiento de datos (*No SQL Data Store*). En lugar de emplear el almacenamiento de una base de datos tradicional, éste ha sido sustituido por

[2] www.jboss.org
[3] www.hibernate.org
[4] http://db.apache.org/derby/

un almacén *no SQL* también llamado almacén clave-valor (*key-value store*). Siendo una tupla de la capa de base de datos un par clave-valor. Actualmente se emplea HBase[5] como almacenamiento de datos no SQL. HBase es una tabla hash distribuida, elástica y replicada de manera automática. Tanto las tablas de datos como los índices de la base de datos son tablas en HBase. HBase se sitúa encima de un sistema de ficheros paralelo-distribuido (*Distributed File System*), Hadoop Distributed File System (HDFS), que constituye la quinta capa y se encarga de proporcionar almacenamiento persistente a los datos.

El uso conjunto de HBase y HDFS[6] permite escalar y replicar el almacenamiento dando lugar a una capa de almacenamiento tolerante a fallos y elástica.

La gestión de transacciones se hace de manera holística y las diferentes capas colaboran para proporcionar propiedades transaccionales [6]. La gestión transaccional depende de un conjunto de componentes que proporcionan distinta funcionalidad: secuenciador de compromisos (*commit sequencers*), servidor de *snapshot* (*snapshot server*), gestores de conflictos (*conflict managers*), y *loggers*. Esta descomposición es crucial para obtener escalabilidad.

Existe una capa adicional en la plataforma encargada de tareas de gestión tales como el despliegue, monitorización, equilibrado de carga dinámico y elasticidad. Cada instancia de cada capa tiene un monitor (*Monitors*) recogiendo datos sobre uso de recursos y mediciones de rendimiento que son notificados a un monitor central que emplea Zookepeer[7], un servidor fiable y distribuido. Este monitor central proporciona estadísticas agregadas al gestor de elasticidad (*Elastic Manager*). El gestor de elasticidad examina los desequilibrios en las instancias de cada capa y reconfigura la capa equilibrando dinámicamente la carga. Si la carga está equilibrada, pero se ha alcanzado el umbral superior de uso de recursos, una nueva instancia se provisiona y es ubicada en esa capa. Si la carga es suficientemente baja para ser satisfecha por un número más pequeño de instancias en una capa, algunas instancias de esa capa transfieren su carga a otras y las instancias que no tienen carga son puestas fuera de servicio.

4. Procesamiento Transaccional Ultra-Escalable

Snapshot isolation [2] es el criterio de corrección empleado para la ejecución concurrente de transacciones en CumuloNimbo. Con *snapshot isolation* (SI) dos transacciones concurrentes tienen conflicto solo si las dos transacciones modifican el mismo dato, es decir, no hay conflictos de lectura-escritura, como ocurre con el criterio de *serialidad*. Esta característica hace que los sistemas de bases de datos que implementan SI escalen más que aquéllos que implementan *serialidad*, ofreciendo garantías muy similares [2].

La escalabilidad de SI frente a serialidad se ha mostrado en el contexto de replicación de bases de datos [8]. En este contexto, los límites de escalabilidad no fueron conflictos de escritura, sino el modelo de replicación, escrituras en todas las réplicas. Por tanto, la carga de escrituras no puede exceder la carga que una réplica puede manejar. CumuloNimbo sólo incluye replicación en la capa de persistencia (HDFS) para tolerar fallos. En el resto de las capas los datos son particionados en distintos nodos de tal forma que cada nodo pueda soportar la carga enviada a los datos que mantiene. Esta división de los datos no es observada por las aplicaciones.

Aunque SI escala mejor que la serialidad, en sistemas que ejecutan varios miles de transacciones por segundo, las tareas de control de concurrencia necesarias para implementar SI pueden convertirse en un cuello de botella. Cuando una transacción ejecuta bajo SI las lecturas se realizan sobre la última versión comprometida en el momento en el que empezó la transacción que lee (*snapshot*). La primera modificación de un dato por parte de la transacción crea una versión privada del dato para esa transacción. Posteriores lecturas por parte de esa transacción del dato modificado se harán sobre esta versión. Si la transacción compromete, sus versiones privadas de los datos modificados se harán visibles. Otras transacciones que empiecen después del compromiso de ésta podrán leerlas. Para implementar la semántica SI, las versiones comprometidas de

[5] hbase.apache.org
[6] hadoop.apache.org/hdfs
[7] http://zookeeper.apache.org/

los datos reciben una marca de tiempo creciente (marca de tiempo de compromiso). Cuando una transacción empieza recibe una marca de tiempo (marca de tiempo de inicio) igual a la marca de tiempo de la última transacción comprometida. Esta transacción leerá la versión de los datos con la mayor marca de tiempo menor o igual que su marca de tiempo de inicio. Los conflictos entre transacciones concurrentes que modifican el mismo dato se pueden manejar de manera optimista (al final de la transacción) o pesimista (empleando cerrojos). Este último método es el que implementan las bases de datos. Tanto la generación de marcas de tiempo de inicio y compromiso como la detección de conflictos se puede convertir en un cuello de botella.

Existen tres componentes principales en el gestor de transacciones de CumuloNimbo para implementar SI: *el gestor de conflictos, el secuenciador de compromisos* y *el servidor de marcas de tiempo*.

El gestor de transacciones de CumuloNimbo proporciona un *gestor distribuido de conflictos*. Cada gestor se ocupa de un conjunto de tuplas. El gestor de conflictos es accedido cada vez que se va a modificar una tupla. Éste comprueba si la tupla ha sido modificada por una transacción concurrente y si es así, aborta la transacción. Si no, anota que la tupla ha sido modificada por la transacción. Existe un gestor de conflictos por cada nodo de caché y éste detecta los conflictos de las tuplas que almacena ese nodo de caché. Ambos están en la misma máquina, de esta forma la detección de conflictos es local.

El *secuenciador de compromisos* proporciona marcas de tiempo de compromiso a las transacciones de actualización cuando comprometen. El *servidor de marcas de tiempo* suministra marcas de tiempo de inicio de transacción.

Hay un gestor de transacciones en cada servidor de aplicaciones. Éste recibe las peticiones de inicio y compromiso de las transacciones. Además, intercepta las operaciones de lectura y escritura. Cuando una transacción empieza, el gestor de transacciones proporciona a la transacción la marca de tiempo de inicio. Esta marca de tiempo es la última marca de tiempo que ha recibido del servidor marcas de tiempo. Las lecturas de esa transacción se harán empleando esa marca de tiempo (marca de tiempo de inicio). Cuando una transacción compromete, se comprueba si ha realizado alguna escritura. Si no es así, se trata de una transacción de lectura que comprometerá localmente. Si la transacción ha modificado datos, el gestor de transacciones local asigna a la transacción y a las tuplas modificadas (writeset) una marca de tiempo de compromiso de las enviada por el secuenciador de compromisos, propaga el writeset tanto al *logger*, encargado de hacer duraderos los cambios, como a la capa de base de datos, la cual lo reenvía a HBase. Cuando el logger notifica que el writeset es duradero, el gestor de transacciones local confirma el compromiso al cliente. Cuando la capa de base de datos confirma que el writeset se ha escrito en HBase, el gestor de transacciones local comunica al servidor de marcas de tiempo que los datos asociados a marca de tiempo de compromiso son visibles y duraderos.

El servidor de marcas de tiempo realiza un seguimiento de qué marcas de tiempo de compromiso han sido usadas y cuáles han sido descartadas. Una marca de tiempo de compromiso se considera usada, cuando el writeset es duradero y visible. El servidor de marcas de tiempo también determina cuál es la marca de tiempo de compromiso más reciente, es decir, aquélla tal que todas las marcas temporales previas hayan sido usadas y/o descartadas, e informa periódicamente al gestor de transacciones local sobre esta marca de tiempo que será empleada por los gestores de transacciones locales como marca de tiempo de inicio. Como las tuplas que tienen esa marca de tiempo (o anteriores) se encuentran en la caché y/o en HBase, las nuevas transacciones leerán los datos más recientes comprometidos (snapshot).

El secuenciador de compromisos es responsable de la asignación de marcas de tiempo de compromiso. Éste envía lotes de marcas de tiempo de compromiso a cada uno de los gestores de transacciones locales en intervalos regulares de tiempo (por ejemplo, cada 10 milisegundos). Para determinar el tamaño adecuado del lote, cada gestor de transacciones informa al servidor de compromiso del número de transacciones de modificación que ha comprometido en el periodo anterior. El tamaño del nuevo lote será función de este valor.

Cuando un gestor de transacciones local recibe un nuevo lote de marcas de tiempo de compromiso, descarta todas las marcas de tiempo no usadas y notifica al servidor marcas de tiempo sobre las mismas, pasando a usar las marcas de tiempo de compromiso del nuevo lote. De esta forma, el

tiempo para realizar el compromiso mínimo. La marca de tiempo de compromiso está en el gestor local de transacciones. La detección de conflictos se realizó cuando la operación de escritura tuvo lugar. El único tiempo de espera ocurre por la escritura del writeset por parte del *logger*. Para evitar un que el *logger* se convierta en un cuello de botella, existen varias instancias del *logger*. Cada una de ellas se encargará de un conjunto de writesets. Las instancias del *logger* se pueden crear dinámicamente, bajo demanda.

5. Resultados preliminares

Se ha construido un prototipo inicial e integrado con JBoss, Hibernate, Derby, HBase y HDFS. El prototipo ha sido evaluado en un cluster de 120 núcleos. El despliegue consta con 5 nodos dual-core para almacenamiento en HBase (*region servers*) y *data nodes* HDFS (ambas instancias en un mismo nodo físico); 1 nodo dual-core para el *Name Node* de HDFS y el aprovisionador de la caché; 1 nodo dual-core para *Zookeeper* y el *Secondary Name Node* de HDFS; 1 nodo dual-core al aprovisionador de servidores de aplicación y al *Master Server* de HBase, 1 nodo dual-core al secuenciador de compromiso; 1 nodo dual-core al servidor marcas de tiempo, 5 nodos dual-core para las instancias de la caché y 5 nodos dual-core para las instancias del *logger*. En un quad-core hay una instancia de JBoss, una instancia de Hibernate y una instancia de Derby. Variamos el número de nodos quad-core dedicados a ellos entre 1, 5 y 20. Con esta configuración se ejecuta el *benchmark* SPECjEnterprise2010[8].

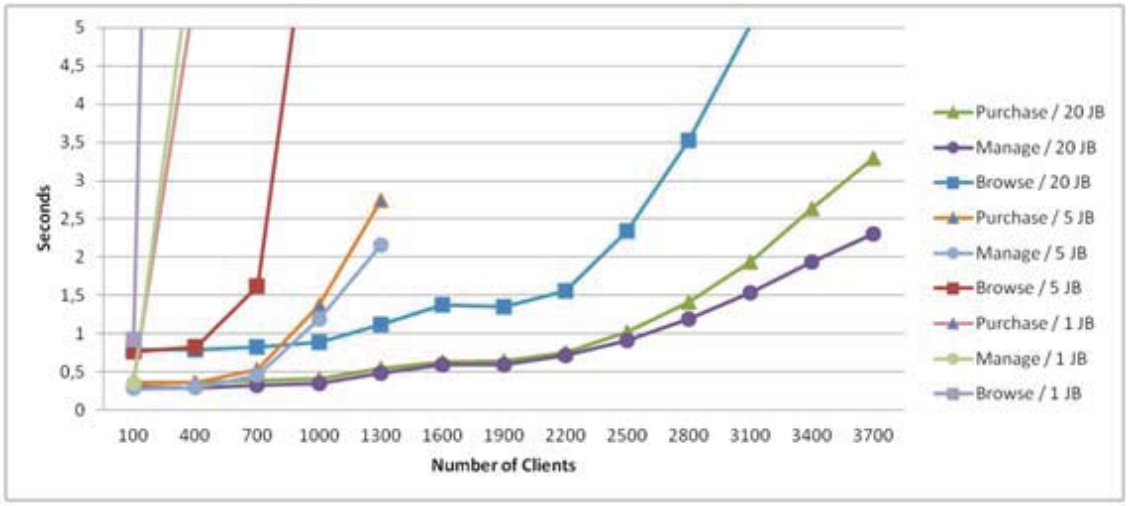


Figura 2. Evolución del tiempo de respuesta con 1 a 20 instancias

La figura 2 muestra los resultados de la evaluación. El umbral del *benchmark* para el tiempo de respuesta es de 2 segundos. Como se puede ver, para las cargas de trabajo con transacciones de escritura (*Manage* y *Purchase*) y con un único nodo JBoss+Hibernate+Derby (Purchase/1 JB y Manage/1 JB) el sistema es capaz de manejar sólo 100 clientes, mientras que con 5 y 20 nodos (5 JB y 20 JB), puede manejar 1000 y entre 3100-3400 clientes. Para la carga de trabajo tipo búsqueda (*Browse*), un nodo JBoss+Hibernate+Derby puede manejar 100 clientes (Browse/1 JB), y con 5 y 20 nodos puede manejar 700 y 2200 clientes (Browse/5 JB y (Browse/20 JB)).

[8] http://www.spec.org/jEnterprise2010/

6. Conclusiones

Este artículo presenta una nueva plataforma transaccional para la *nube*, CumuloNimbo. Las principales características de CumuloNimbo son: trasacciones ACID sin necesidad de particionar los datos (*sharding*), coherencia en todas las capas, transparencia completa, sintáctica y semánticamente. Los resultados preliminares muestran que el sistema escala. El principal problema de rendimiento que hemos encontrado está relacionado con la falta de concurrencia de HBase y el equilibrado de carga que realiza.

Referencias

1. J. Baker, C. Bond, J. Corbett, J. J. Furman, A. Khorlin, J. Larson, J.-M. Leon, Y. Li, A. Lloyd, and V. Yushprakh. Megastore: Providing scalable, highly available storage for interactive services. In *CIDR*, pages 223–234, 2011.
2. H. Berenson, P. Bernstein, J. Gray, J. Melton, E. J. O'Neil, and P. E. O'Neil. A critique of ansi sql isolation levels. In *SIGMOD*, 2005.
3. P. Bernstein, C. W. Reid, and X. Y. M. Wu. Optimistic concurrency control by melding trees. In *Int. Conf. on Very Large Data Bases.*, 2011.
4. M. Brantner, D. Florescu, D. A. Graf, D. Kossmann, and T. Kraska. Building a database on s3. In *SIGMOD Conference*, pages 251–264, 2008.
5. S. Das, D. Agrawal, and A. E. Abbadi. Elastras: An elastic transactional data store in the cloud. *CoRR*, abs/1008.3751, 2010.
6. R. Jiménez-Peris and M. Patiño-Martínez. *System and Method for Highly Scalable Decentralized and Low Contention Transactional Processing.*
7. R. Jiménez-Peris, M. Patiño-Martínez, I. Brondino, B. Kemme, R. Oliveira, J. Pereira, R. Vilaa, F. Cruz, and Y. Ahmad. CumuloNimbo: Parallel-distributed transactional processing. In *Cloud Futures*, 2012.
8. B. Kemme, R. Jimenez-Peris, and M. Patiño-Martinez. *Database Replication.* Morgan & Claypool Publishers, 2010.
9. J. J. Levandoski, D. B. Lomet, M. F. Mokbel, and K. Zhao. Deuteronomy: Transaction support for cloud data. In *CIDR*, pages 123–133, 2011.
10. Microsoft.com. Microsoft SQL Azure Database. Submitted for publication, 2010. http://www.microsoft.com/azure/data.mspx.
11. D. Peng and F. Dabek. Large-scale incremental processing using distributed transactions and notifications. In *Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2010.
12. Z. Wei, G. Pierre, and C.-H. Chi. CloudTPS: Scalable transactions for Web applications in the cloud. *IEEE Transactions on Services Computing*, 2011.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): Actas de las "VIII Jornadas de Ciencia e Ingeniería de Servicios (JCIS'2012)", Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

Sesión 2

Ingeniería de Servicios

Chair: *Dr. Pedro Alvarez*

Sesión 2: Ingeniería de Servicios
Chair: Dr. Pedro Alvarez

Antonio García Domínguez and Inmaculada Medina Bulo. *Un Método de Generación de Pruebas de Rendimiento para Múltiples Tecnologías desde Modelos UML con Anotaciones MARTE*.

Juan Carlos Castillo Cano, Francisco Almeida, Vicente Blanco and María Carmen Ramírez Castillejo. *Plataforma de computación genérica basada en servicios web para problemas de conteo de células*.

Jorge Moratalla and Esperanza Marcos. *Definición y Aplicación de un proceso de Modernización y Evolución al Sistema de Gestión de Nombres de Dominios ".es"*.

Miguel A. González-Serrano, Diana Perez-Marin and Miren Idoia Alarcón. *Clasificación de los Servicios Web de Negocio Corporativos basada en la Funcionalidad Horizontal de las Organizaciones*.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): Actas de las "VIII Jornadas de Ciencia e Ingeniería de Servicios (JCIS'2012)", Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

Un Método de Generación de Pruebas de Rendimiento para Múltiples Tecnologías desde Modelos UML con Anotaciones MARTE

Antonio García Domínguez[1] e Inmaculada Medina Bulo[2]

[1] Universidad de Cádiz, Escuela Superior de Ingeniería
C/Chile 1, CP 11002, Cádiz, España,
antonio.garciadominguez@uca.es,
Sitio web: http://neptuno.uca.es/~agarcia
[2] Universidad de Cádiz, Escuela Superior de Ingeniería
C/Chile 1, CP 11002, Cádiz, España,
inmaculada.medina@uca.es,
Sitio web: http://neptuno.uca.es/~imedina

Resumen Obtener el rendimiento esperado de un flujo de trabajo sería más fácil si cada tarea incluyera sus propias especificaciones. Sin embargo, normalmente sólo se dan requisitos globales de rendimiento, obligando a los diseñadores a inferir los requisitos locales a mano. En trabajos anteriores presentamos dos algoritmos que automáticamente inferían restricciones locales de rendimiento a partir de diagramas de actividad Unified Modelling Language anotados mediante el perfil Modelling and Analysis of Real-Time and Embedded Systems. En este trabajo presentamos un método para usar estas anotaciones para generar casos de prueba de rendimiento para múltiples tecnologías, relacionando el modelo de rendimiento con modelos de diseño e implementación. Mostramos cómo se podría aplicar a código Java y a composiciones de servicios mediante tecnologías existentes de código abierto, y estudiamos las tareas a realizar para su implementación y las similitudes y diferencias con otras propuestas.

Palabras clave: rendimiento del software, Servicios Web, ingeniería dirigida por modelos, MARTE, UML

1. Introducción

Todo software ha de cumplir requisitos funcionales y no funcionales. Los requisitos de rendimiento son unos de los requisitos no funcionales más comúnmente dados, y en algunos contextos pueden ser tan importantes como los requisitos funcionales. Son comunes no sólo en sistemas de tiempo real duro o blando, sino también en las Arquitecturas Orientadas a Servicios («Service Oriented Architectures» o SOA en inglés). Trabajando con las SOA, es muy común establecer Acuerdos de Nivel de Servicio («Service Level Agreements» o SLA) con los proveedores externos, garantizando el nivel mínimo de rendimiento que se necesita.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 51-64, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

Otra tarea muy común en las SOA es crear «composiciones de servicios» [9]. Son servicios que reúnen a otra serie de servicios de nivel inferior en uno con mayor funcionalidad. En las composiciones de servicios resulta difícil cuantificar los requisitos de los SLA de los servicios integrados. Si se pide demasiado, puede que se incurra en costes adicionales. Si se pide muy poco, puede que no se consiga el nivel de calidad deseado. Además, en caso de que no se cumpla este nivel de calidad será necesario buscar la causa raíz del problema. Tener requisitos más localizados de rendimiento agilizaría el proceso.

En nuestro trabajo anterior [10] presentamos dos algoritmos de inferencia de restricciones de rendimiento en modelos de flujos de trabajo. Estos algoritmos pueden completar un modelo con los tiempos de respuesta a exigir bajo ciertas cargas, partiendo de una anotación global y algunas anotaciones locales opcionales. Los usuarios podrían simplemente tomar estos modelos completados y usarlos para derivar manualmente pruebas de rendimiento. Sin embargo, escribir estas pruebas para cada parte de un sistema de cierto tamaño podría ser bastante costoso. Lo ideal sería automatizarlo en la medida de lo posible.

En este trabajo presentamos una forma de reducir el esfuerzo necesario para elaborar estas pruebas, ayudando al usuario a sacar el máximo partido a los modelos de rendimiento obtenidos. A partir de los modelos se generarán planes parciales de pruebas y se envolverán pruebas funcionales existentes como pruebas de rendimiento. Para ello, combinaremos los modelos de rendimiento con modelos de diseño y/o implementación, relacionando los requisitos de rendimiento con los artefactos software oportunos.

Es importante recalcar que nuestro uso de «prueba de rendimiento» es más específico que el usado por ISO, IEC e IEEE. Según [12], «prueba de rendimiento» («performance requirement») puede entenderse como «the measurable criterion that identifies a quality attribute of a function or how well a functional requirement must be accomplished». Esta definición es muy abierta, dejando como «rendimiento» el nivel de calidad obtenido en cualquier tipo de atributo. En este trabajo nos restringiremos a dos de los atributos de calidad (en particular «throughput» o productividad y «response time» o tiempo de respuesta) medibles mediante un análisis de rendimiento («performance analysis»), que según [12] consiste en «a qualitative analysis of a real-time system (or software design) executing on a given hardware configuration with a given external workload applied to it». Por lo tanto, las pruebas de rendimiento nos exigen definir unos objetivos de rendimiento bajo una carga de trabajo determinada.

El resto de este trabajo se estructura de la siguiente forma: tras introducir los modelos usados, describiremos nuestro método general para generar pruebas. Mostraremos dos formas de aplicar este enfoque, relacionando los requisitos de rendimiento con varios tipos de artefactos software, y seleccionaremos las tecnologías a adoptar. Finalmente, cerraremos este trabajo con una serie de conclusiones y nuestras futuras líneas de trabajo.

2. Modelos de rendimiento

En esta sección presentamos la notación utilizada por los algoritmos descritos en [10]. Utilizamos diagramas de actividad UML anotados con un subconjunto reducido del perfil «Modelling and Analysis of Real-Time and Embedded Systems» (MARTE) [16]. MARTE proporciona tanto un conjunto de métricas predefinidas de rendimiento, como un marco para crear nuevas métricas. En particular, usamos algunas de las métricas predefinidas del subperfil «Generic Quantative Analysis Modelling» (GQAM). GQAM es el subperfil básico para realizar análisis de rendimiento: otros subperfiles como «Schedulability Analysis Modelling» (SAM) y «Performance Analysis Modelling» (PAM) se basan en él. Sin embargo, no utilizamos ni SAM ni PAM.

La figura 1 muestra un ejemplo sencillo de nuestra notación. Las anotaciones inferidas se destacan mediante negritas:

1. La actividad está anotada mediante el estereotipo ≪GaScenario≫. El atributo `respT` indica que cada petición debe ser atendida en menos de un segundo, y `throughput` indica que se recibirá al menos 1 petición por segundo.
2. Cada acción en la actividad se anota con ≪GaStep≫. El campo `hostDemand` incluye una expresión de la forma $m+ws$, donde m es el mínimo tiempo que ha de recibir, w es el peso a la hora de distribuir el tiempo sobrante, y s es el parámetro de contexto relacionado con la acción.
   El algoritmo de inferencia de tiempos límite añade una nueva restricción a `hostDemand`, indicando el tiempo límite concreto que se ha de respetar. El algoritmo de inferencia de peticiones por segundo extiende `throughput` con una restricción que especifica cuántas peticiones por segundo se deberían atender. Para indicar que estas restricciones se han calculado automáticamente, utilizamos sus campos `source`, que se establecen a `calc` (calculado).
   Se pueden modelar diversas situaciones según la forma en que se combinen los valores de m y w. Por ejemplo, «Evaluar Pedido» tiene $m = 0.4$ y $w = 0$, indicando que se tiene un límite fijo de tiempo de 0.4 s, proveniente de un SLA ya existente. Sin embargo, el resto de acciones tienen $m = 0$ y $w = 1$, indicando que no tenemos SLA firmados ni mediciones de tiempos de ejecución, con lo que sus tiempos límite estarán fijados completamente por el algoritmo. Además, como todos tienen el mismo peso, se ha estimado que tienen aproximadamente el mismo coste de ejecución.
3. La actividad también declara una serie de parámetros de contexto en el campo `contextParam` del estereotipo ≪GaAnalysisContext≫. Estas variables representan el tiempo por unidad de peso que ha de recibir su actividad correspondiente, más allá del tiempo mínimo que solicitó. El algoritmo de inferencia de tiempos límite se ocupa de calcular sus valores.
   En el diagrama hemos adoptado la convención de utilizar «sw» («Slack per unit of Weight») seguido de las iniciales del nombre de la acción. De esta forma, `swCR` indica que «Crear Recibo» debería recibir 0.2 segundos por unidad de peso. Dado que su tiempo mínimo es 0 y tiene peso 1, el tiempo límite final será de $0 + 1 \cdot 0.2 = 0.2$ segundos.

4. Las aristas salientes de los nodos condición también usan ≪GaStep≫, pero sólo emplean el campo `prob`. El usuario ha de indicar en este campo cuál es la probabilidad estimada de que esa rama sea cierta, frente al resto de sus ramas hermanas.
   En la figura 1 se han anotado las aristas del único nodo decisión con probabilidades, indicando que el 80 % de los pedidos son aceptados y el 20 % son rechazados.

3. Método general

El modelo mostrado en la sección anterior es completamente abstracto. Con su nivel de detalle no se puede ejecutar de forma automática. El desarrollador deberá implementar la composición por otros medios, ya sea escribiendo el código a mano o generándolo usando un enfoque dirigido por modelos.

Una vez implementado, sería útil poder aprovechar el modelo original para generar las pruebas de rendimiento. Sin embargo, el modelo no tiene los detalles necesarios para producir artefactos ejecutables. Para resolver este problema, se pueden considerar varias alternativas. Una opción sería extender el modelo de rendimiento o el modelo de diseño con información del otro modelo, pero esto los complicaría con detalles no relacionados con su propósito original. Para evitar esta pérdida de cohesión de los modelos, sería mejor utilizar un modelo aparte para reunir a los modelos anteriores. Dichos modelos se suelen llamar modelos *de entrelazado* o «weaving models». Existen diversas tecnologías para implementarlos, como AMW [7] o ModeLink [15]. Esta es la opción que hemos escogido para nuestro método.

Tras crear las relaciones oportunas, el siguiente paso será generar las pruebas en sí. Se podría utilizar una transformación de modelo a texto (Model to Text o M2T) normal y corriente, escrita en un lenguaje especializado como el Epsilon Generation Language [14]. Sin embargo, es posible que haga falta refinar ligeramente el modelo intermedio antes de aplicar esta transformación. En dicho caso, se podría añadir una transformación de modelo a modelo (Model to Model o M2M) intermedia. La figura 2 resume nuestro enfoque de forma gráfica.

4. Aplicaciones

En esta sección mostramos varias formas de aplicar el enfoque resumido en la figura 2, usando diversas tecnologías para ayudar a generar artefactos de pruebas de rendimiento en diversos entornos.

4.1. Reutilización de pruebas funcionales

Generar casos de prueba ejecutables desde cero automáticamente requeriría muchos modelos detallados y transformaciones complejas que serían costosos de elaborar y mantener. Una alternativa más simple y barata sería reutilizar

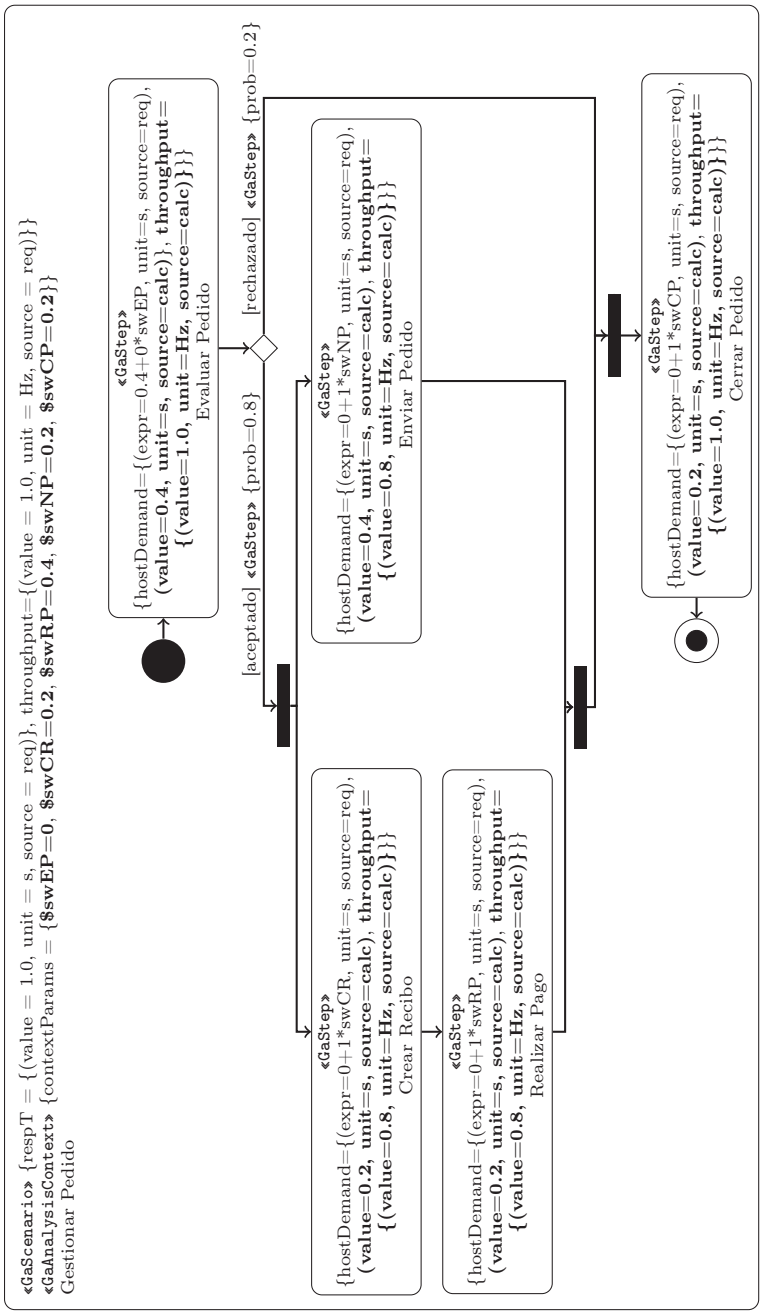


Figura 1. Ejemplo sencillo de un modelo anotado por los algoritmos de rendimiento

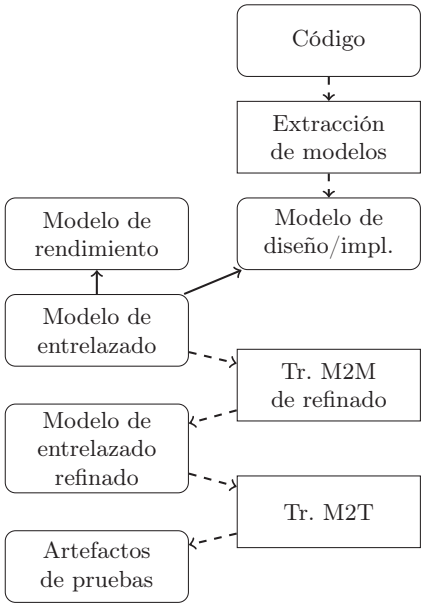


Figura 2. Método general para generar artefactos de pruebas de rendimiento a partir de modelos abstractos de rendimiento

pruebas funcionales existentes como pruebas de rendimiento. Se pueden utilizar varias bibliotecas existentes para ello, como ContiPerf [4] o JUnitPerf [6]. El código a generar en este caso deberá utilizar alguna de estas bibliotecas, para que la transformación sea más sencilla y el código sea más legible.

El listado 1 muestra un ejemplo de uso de JUnitPerf. El caso de prueba funcional original *TFuncional* se envuelve mediante un objeto de la clase *TimedTest* (proporcionada por JUnitPerf), comprobando que cada caso de prueba no tarde más de 1000 milisegundos. A continuación se envuelve de nuevo con *LoadTest* (también de JUnitPerf), emulando a 10 usuarios ejecutando la prueba al mismo tiempo. Combinando estos dos envoltorios, la prueba resultante comprueba que cada una de las 10 ejecuciones concurrentes de la prueba tarda menos de 1 segundo.

El listado 2 muestra código equivalente al anterior, utilizando ContiPerf. En vez de envolver las pruebas mediante objetos, ContiPerf emplea anotaciones de Java 6, que son más fáciles de generar automáticamente. La anotación *@PerfTest* indica que la prueba se ejecutará 100 veces usando 10 hilos, de forma de que cada hilo realice 10 llamadas. *@Required* indica que cada llamada debería completarse en 1000 milisegundos como mucho. *@SuiteClasses* lista los conjuntos de casos de prueba JUnit 4 que deberían reutilizarse para hacer pruebas de rendimiento, y *@RunWith* integra a ContiPerf con el marco de pruebas unitarias JUnit 4.

En ambos casos, el código es muy sencillo. Sin embargo, el código generado debe integrarse correctamente con el código ya existente. Si el código no se

```
final int usuarios = 10;
final int limite_ms = 1000;
Test caso = new TFunctional();
Test caso1Usuario = new TimedTest(caso, limite_ms);
Test casoNUsuarios = new LoadTest(caso1Usuario, usuarios);
```

Listado 1. Código Java que envuelve la prueba funcional *TFunctional* de JUnit 3 mediante JUnitPerf

```
@RunWith(ContiPerfSuiteRunner.class)
@SuiteClasses(TFunctionalJUnit4.class)
@PerfTest(invocations = 100, threads = 10)
@Required(max=1000)
public class TRendimiento {}
```

Listado 2. Código Java que decora el conjunto de pruebas para JUnit 4 *TFunctional-JUnit4* mediante ContiPerf

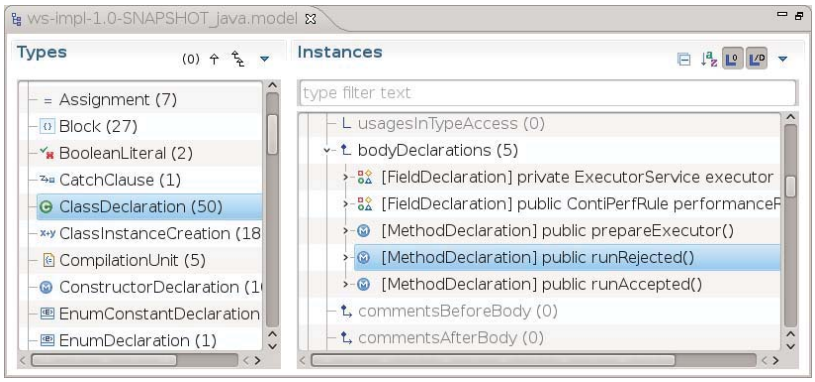


Figura 3. Toma de pantalla del navegador de modelos de MoDisco, mostrando un modelo generado a partir de un proyecto Java de Eclipse

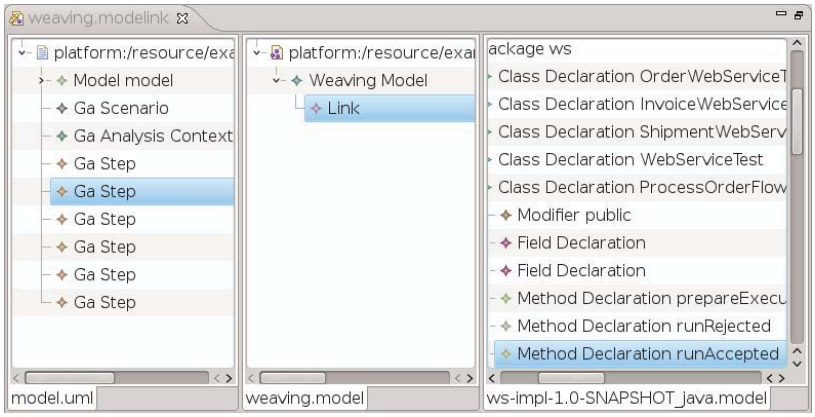


Figura 4. Toma de pantalla de Epsilon ModeLink, relacionando dos modelos EMF

```
@WebService public class HolaMundo {
  @WebMethod public String greet(@WebParam(name="nombre") String nombre) {
    return "Hola␣" + nombre;
  }
}
```

Listado 3. Código Java que implementa el servicio «HolaMundo» mediante JAX-WS

produjo utilizando un enfoque dirigido por modelos, no existirán modelos de diseño o implementación que relacionar con el modelo de rendimiento. En su lugar, tendremos que extraer un modelo del código, usando una herramienta como Eclipse MoDisco [5]. Eclipse MoDisco puede generar modelos a partir de código Java, como el mostrado en la figura 3. En esta figura se ha seleccionado un objeto *MethodDeclaration* que representa a un caso de prueba JUnit 4. El usuario vincularía un requisito de rendimiento a este objeto en el modelo de entrelazado.

Tras enlazar el modelo de rendimiento con el modelo de diseño o implementación empleando Epsilon ModeLink (véase la figura 4), el último paso será lanzar la transformación M2T para generar los artefactos de pruebas. El código generado será parecido al de los listados 1 o 2.

4.2. Generación de planes parciales de pruebas para Servicios Web

En la sección anterior aplicamos nuestro enfoque sobre pruebas unitarias JUnit ya existentes, convirtiéndolas en pruebas de rendimiento. En esta sección describiremos cómo generar pruebas de rendimiento para un Servicio Web (Web Service o WS) [11] de forma independiente del lenguaje de implementación.

Las interfaces públicas de los Servicios Web basados en la pila de tecnologías WS-* normalmente se describen mediante documentos en el Web Services

Description Language (WSDL) [24]. Este lenguaje describe las operaciones disponibles y los mensajes implicados de forma independiente del lenguaje de programación usado para implementar el servicio. Muchos de los marcos existentes para implementar Servicios Web (como Apache CXF [1]) pueden generar gran parte del código necesario a partir de un documento WSDL, de forma que los usuarios sólo tengan que completarlo con la lógica de negocio oportuna. Además, algunos marcos (incluyendo a CXF) pueden operar al revés, generando documentos WSDL a partir de código que haya sido debidamente anotado.

El listado 3 muestra un fragmento de código Java de ejemplo que implementa un Servicio Web «Hola Mundo» usando anotaciones JAX-WS [13] estándar. Se podría pensar en utilizar el mismo enfoque de la sección 4.1 para el servicio del listado 3, ya que es código Java normal y corriente. Sin embargo, si el servicio se escribiera en otro lenguaje, habría que adaptar el proceso de generación de código a él.

Dado que un documento WSDL es precisamente una descripción abstracta y declarativa del Servicio Web, lo podemos utilizar como modelo de diseño. Tras transformar automáticamente la descripción XML Schema del formato WSDL a un metamodelo ECore [18], podremos cargar documentos WSDL como modelos del Eclipse Modelling Framework (EMF), y podremos aprovechar la mayoría de las tecnologías mencionadas en la sección 4.1.

El modelo de entrelazado ha de relacionar los estereotipos ≪GaStep≫ de las acciones con las operaciones de los servicios en el documento WSDL. Por ejemplo, puede que necesitemos asegurarnos de que cada llamada a la operación *evaluar* del servicio de *Pedidos* termine en un cierto tiempo bajo una demanda de un cierto número de peticiones por segundo.

Una vez se establezcan estas relaciones entre el modelo WSDL y el modelo de rendimiento, el siguiente paso será generar un plan de pruebas para una herramienta especializada de pruebas de rendimiento, como por ejemplo The Grinder [2]. Usar una herramienta especializada permite definir las pruebas de forma más sencilla y de manera independiente al lenguaje de implementación del Servicio Web.

En el caso específico de The Grinder, habrá que generar dos archivos: un fichero `.properties` especificando la carga a generar, y un guión Jython con la prueba que cada cliente emulado ha de ejecutar. El fichero `.properties` del listado 4 indica que 5 procesos deberían ejecutar la prueba 100 veces cada uno, y que se debería empezar con 1 proceso y añadir uno más cada 1000 milisegundos. Por otro lado, la prueba del listado 5 envía un mensaje apropiado al servicio Web y comprueba que la respuesta tiene el código de estado HTTP «OK» (200) y que se recibe en menos de 150 milisegundos. Dado que estos ficheros son muy concisos, creemos que es factible generar una versión inicial de ambos. El usuario sólo tendrá que completarlos con un mensaje SOAP apropiado.

Esta aplicación concreta del enfoque se podría refinar posteriormente para ayudar al usuario a generar los mensajes. También se podrían utilizar los enlaces del modelo de entrelazado para especificar una estrategia concreta de generación. Por ejemplo, se podrían generar aleatoriamente, realizar variaciones sobre una

```
grinder.processes=5
grinder.runs=100
grinder.processIncrement=1
grinder.processIncrementInterval=1000
```

Listado 4. Fichero `.properties` de ejemplo con la configuración de la carga

```
class TestRunner:
  def __call__(self):
    def invoke():
      respuesta = HTTPRequest().POST("http://localhost:8080/pedidos",
                                     " (... mensaje_SOAP ...)")
      stats = grinder.statistics.getForCurrentTest()
      stats.success = respuesta.statusCode != 200 and stats.time < 150
      test = Test(1, "Consultar pedido por ID").wrap(invoke)
      test()
```

Listado 5. Guión Jython de ejemplo para The Grinder con la prueba de rendimiento a ejecutar por cada cliente simulado

plantilla o aplicar análisis estático del código del Servicio Web. La estrategia se podría aplicar en el paso de refinamiento del modelo de entrelazado de la figura 2.

5. Trabajos relacionados

Según Woodside et al. [23], la ingeniería de rendimiento abarca todas las actividades necesarias para cumplir los requisitos de rendimiento de un sistema. Algunas de estas actividades son definir los requisitos, analizar modelos abstractos de rendimiento (como redes de colas en capas [17] o especificaciones en álgebra de procesos [20]) o probar el rendimiento del sistema implementado. Nuestro trabajo anterior en [10] se centró en ayudar al usuario a definir los requisitos utilizando diagramas de actividad UML con anotaciones MARTE [16]. Este trabajo se ocupa de aprovechar esos modelos para crear los artefactos de pruebas de rendimiento.

Nuestro enfoque no manipula directamente el sistema implementado, sino una representación simplificada (un *modelo*). Existe un gran número de trabajos que tratan sobre pruebas dirigidas por modelos o «model-based testing». Según la revisión del estado del arte de Utting et al. en [21], consiste en la «derivación automatizable de casos concretos de prueba a partir de modelos formales abstractos, y su ejecución». La mayoría de los trabajos en la literatura se dedican a las pruebas funcionales: nos centraremos en aquellos trabajos dedicados a las pruebas de rendimiento.

Barna et al. presentan en [3] un enfoque híbrido que emplea una red de colas en dos capas para derivar automáticamente una carga inicial para un sitio web.

Esta carga inicial se utiliza para probar el sistema y refinar el modelo original iterativamente, buscando una carga mínima que haga que el sistema viole uno de sus requisitos de rendimiento. De forma similar a nuestro trabajo, combina el análisis de un modelo con la ejecución de una serie de casos de prueba. Sin embargo, su meta es completamente distinta a la nuestra. Nuestro objetivo es definir los niveles de servicio a exigir a los servicios integrados de forma que se cumpla el nivel de servicio global. El trabajo anterior sólo estimaría la carga máxima que el servicio existente puede soportar con los niveles deseados de servicio.

Di Penta et al. han publicado otro enfoque con la misma meta de encontrar cargas de trabajo que hagan que se incumplan los niveles de calidad de servicio exigidos [8]. Sin embargo, utilizan algoritmos genéticos en vez de redes de colas en capas, y prueban Servicios Web basados en WSDL en vez de un sitio web estándar.

Suzuki et al. han desarrollado un enfoque dirigido por modelos para generar entornos de pruebas para Servicios Web [19]. Emplean los niveles de servicio y varios modelos de comportamiento para emular los servicios externos integrados por una composición. De esta forma, los usuarios pueden probar que siempre que se cumplan los niveles de servicio estipulados, se cumplirá el nivel de servicio global deseado. Sin embargo, este enfoque no genera los mensajes de entrada para los servicios en sí. De todos modos, podríamos utilizar este trabajo para comprobar la validez de las restricciones inferidas por nuestros algoritmos.

A partir de las referencias anteriores, se puede deducir que existe una amplia variedad de métodos para generar pruebas de rendimiento y entornos de pruebas para Servicios Web. Sin embargo, no hemos podido encontrar otro uso de entrelazado de modelos para generar artefactos de pruebas de rendimiento para múltiples tecnologías, a pesar de que el entrelazado de modelos se ha usado considerablemente desde la publicación del ATLAS Model Weaver [7]. Por ejemplo, Vara et al. utilizan composición de modelos para decorar sus modelos de casos de uso extendidos con información adicional para una transformación posterior [22].

6. Conclusiones y trabajo futuro

En este trabajo hemos descrito un método general para generar artefactos para pruebas de rendimiento a partir de los modelos abstractos de rendimiento producidos por los algoritmos de inferencia de nuestro trabajo anterior en [10]. Para generar artefactos concretos sin complicar los modelos de rendimiento con detalles de implementación, proponemos relacionar el modelo de rendimiento con un modelo de diseño o implementación a través de un modelo intermedio de entrelazado («weaving model»). Si un modelo de diseño o implementación no está disponible, se puede extraer del código existente. Antes de generar los artefactos, puede que sea necesario refinar el modelo de entrelazado con una transformación de modelo a modelo previa.

Hemos realizado un estudio inicial de cómo se podría aplicar nuestro enfoque en dos situaciones concretas. El primer caso de estudio reutilizará pruebas unitarias en JUnit como pruebas de rendimiento, empleando JUnitPerf o ContiPerf. El modelo de implementación se extraerá del código Java de las pruebas utilizando la herramienta de extracción de modelos MoDisco [5], y el modelo de entrelazado reunirá las anotaciones MARTE de nuestros modelos de rendimiento con las pruebas JUnit del modelo de MoDisco.

El segundo caso de estudio generará planes de pruebas para The Grinder [2], una herramienta especializada de pruebas de rendimiento independiente del lenguaje de implementación del sistema bajo prueba. En este caso, la descripción WSDL de la interfaz del servicio servirá como modelo de diseño, y las anotaciones MARTE se relacionarán con una operación concreta del servicio. En versiones posteriores se considerará aprovechar el modelo de entrelazado para especificar una estrategia de generación para los mensajes de entrada.

El siguiente paso es implementar las transformaciones exigidas por estos enfoques. Una parte considerable del primer caso de estudio ya se encuentra implementada por MoDisco y Epsilon ModeLink. Actualmente, estamos implementando la generación del código de las pruebas utilizando el Epsilon Generation Language.

Agradecimientos

Este trabajo fue financiado por la beca de investigación PU-EPIF-FPI-C 2010-065 de la Universidad de Cádiz, por el proyecto MoDSOA (TIN2011-27242) del Programa Nacional de Investigación, Desarrollo e Innovación del Ministerio de Ciencia e Innovación y por el proyecto PR2011-004 del Plan de Promoción de la Investigación de la Universidad de Cádiz.

Referencias

1. Apache Software Foundation: Apache CXF (noviembre 2011), `https://cxf.apache.org/`, última comprobación: 2012-04-04.
2. Aston, P., Fizgerald, C.: The Grinder, a Java Load Testing Framework (2012), `http://grinder.sourceforge.net/`, última comprobación: 2012-04-04.
3. Barna, C., Litoiu, M., Ghanbari, H.: Model-based performance testing (NIER track). In: Proceedings of the 33rd International Conference on Software Engineering. pp. 872–875. ICSE '11, ACM, Nueva York, NY, EEUU (2011)
4. Bergmann, V.: ContiPerf 2 (septiembre 2011), `http://databene.org/contiperf.html`, última comprobación: 2012-04-04.
5. Bruneliere, H., Cabot, J., Jouault, F., Madiot, F.: MoDisco: a generic and extensible framework for model driven reverse engineering. In: Proceedings of the IEEE/ACM International Conference on Automated Software Engineering. pp. 173–174. Antwerp, Bélgica (septiembre 2010)
6. Clark, M.: JUnitPerf (octubre 2009), `http://clarkware.com/software/JUnitPerf.html`, última comprobación: 2012-04-04.

7. Del Fabro, M.D., Bézivin, J., Valduriez, P.: Weaving models with the Eclipse AMW plugin. In: Proceedings of the 2006 Eclipse Modeling Symposium, Eclipse Summit Europe. Esslingen, Alemania (octubre 2006)
8. Di Penta, M., Canfora, G., Esposito, G., Mazza, V., Bruno, M.: Search-based testing of service level agreements. In: Lipson, H. (ed.) Proceedings of Genetic and Evolutionary Computation Conference. pp. 1090–1097. ACM, Londres, Reino Unido (julio 2007)
9. Erl, T.: SOA: Principles of Service Design. Prentice Hall, Indiana, EEUU (2008)
10. García-Domínguez, A., Medina-Bulo, I.: Inferencia automática de requisitos locales de rendimiento en flujos de trabajo anotados con MARTE. In: Actas de las XVI Jornadas de Ingeniería del Software y Bases de Datos. pp. 585–598. Servizo de publicacións da Universidade da Coruña, A Coruña, España (septiembre 2011)
11. Haas, H., Brown, A.: Web services glossary. W3C working group note, World Wide Web Consortium (febrero 2004), http://www.w3.org/TR/ws-gloss/, última comprobación: 2012-04-04.
12. IEEE, ISO, IEC: 24765-2010: Systems and software engineering — vocabulary. Tech. rep. (marzo 2011), http://dx.doi.org/10.1109/IEEESTD.2010.5733835
13. Java.net: JAX-WS reference implementation (noviembre 2011), http://jax-ws.java.net/, última comprobación: 2012-04-04.
14. Kolovos, D.S., Paige, R.F., Rose, L.M., García-Domínguez, A.: The Epsilon Book (2011), http://www.eclipse.org/epsilon/doc/book, última comprobación: 2012-04-04.
15. Kolovos, D.S.: Epsilon ModeLink (2010), http://eclipse.org/epsilon/doc/modelink/, última comprobación: 2012-04-04.
16. Object Management Group: UML Profile for Modeling and Analysis of Real-Time and Embedded systems (MARTE) 1.0 (noviembre 2009), http://www.omg.org/spec/MARTE/1.0/, última comprobación: 2012-04-04.
17. Petriu, D.C., Shen, H.: Applying the UML Performance Profile: Graph Grammar-based Derivation of LQN Models from UML Specifications. In: Proc. of the 12th Int. Conference on Computer Performance Evaluation: Modelling Techniques and Tools (TOOLS 2002), Lecture Notes in Computer Science, vol. 2324, pp. 159—177. Springer Berlin, Londres, Reino Unido (2002)
18. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: EMF: Eclipse Modeling Framework. Eclipse Series, Addison-Wesley Professional, segunda edn. (diciembre 2008)
19. Suzuki, K., Higashino, T., Ulrich, A., Hasegawa, T., Bertolino, A., De Angelis, G., Frantzen, L., Polini, A.: Model-based generation of testbeds for web services. In: Testing of Software and Communicating Systems, Lecture Notes in Computer Science, vol. 5047, pp. 266–282. Springer Berlin Heidelberg (2008)
20. Tribastone, M., Gilmore, S.: Automatic extraction of PEPA performance models from UML activity diagrams annotated with the MARTE profile. In: Proc. of the 7th Int. Workshop on Software and Performance. pp. 67–78. ACM, Princeton, NJ, EEUU (2008)
21. Utting, M., Pretschner, A., Legeard, B.: A taxonomy of model-based testing. Working Paper 04/2006 (abril 2006), http://researchcommons.waikato.ac.nz/handle/10289/81, última comprobación: 2012-04-04.
22. Vara, J.M., De Castro, M.V., Didonet Del Fabro, M., Marcos, E.: Using weaving models to automate model-driven web engineering proposals. International Journal of Computer Applications in Technology 39(4), 245–252 (2010)
23. Woodside, M., Franks, G., Petriu, D.: The future of software performance engineering. In: Proc. of Future of Software Engineering 2007. pp. 171–187 (2007)

24. World Wide Web Consortium: WSDL 2.0 part 1: Core Language. http://www.w3.org/TR/wsdl20 (junio 2007), última comprobación: 2012-04-04

PLATAFORMA DE COMPUTACIÓN GENÉRICA BASADA EN SERVICIOS WEB PARA PROBLEMAS DE CONTEO DE CÉLULAS

J. C. Castillo, F. Almeida, V. Blanco, M.C. Ramírez

Departamento de Estadística, Investigación Operativa y Computación, Universidad de la Laguna, España
Labotario de Biología Celular y Molecular de C.M., Universidad de Castilla la Mancha, España
{jcastill, falmeida, vblanco}@ull.es, Carmen.Ramirez@uclm.es

Resumen El procesado y análisis de imágenes de células es una tarea vital en los laboratorios de cualquier campo de las ciencias de la salud. Obtener información de esas imágenes, como el número de células que cumplan una serie de requerimientos, es una tarea muy extendida que se realiza principalmente por técnicos de laboratorio, apoyados frecuentemente por algún tipo de software adaptado a los requerimientos del experimento. En este trabajo planteamos el desarrollo de una plataforma basada en servicios web orientada a problemas de conteo de células. Haciendo uso del framework de desarrollo para servicios web OpenCF integramos en un mismo entorno servicios orientados al procesado y clasificación de la imagen, el conteo de las células que cumplan una serie de parámetros y el post-procesado de los datos (generación de gráficas, hojas de datos, etc). Añadimos a dicha plataforma una interfaz gráfica para poder lanzar procesos con conjuntos de imágenes, así como servicios web para la ejecución de las distintas tareas desde un cliente orientado a servicios web.

1. Introdución

El reconocimiento y la búsqueda de patrones en imágenes es un problema que aparece frecuentemente asociado a la investigación de diversas disciplinas. El problema está siendo ampliamente estudiado y se encuentra en constante evolución. En el ámbito de las ciencias de la salud, identificar patrones y realizar recuentos de microorganismos o células presentes en una muestra es uno de los trabajos que deben realizarse a diario en los laboratorios. Este trabajo es desarrollado principalmente por personal técnico cualificado, utilizando herramientas que pueden variar desde simples microscopios hasta potentes equipos de análisis. El proceso suele ser costoso en tiempo y recursos, siendo un escollo importante en la obtención de resultados de procesos analíticos o investigaciones médicas.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 65-76, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

El procedimiento de identificación y cálculo del número de células en el ámbito de las ciencias de la salud suele seguir un mismo patrón en la mayoría de situaciones, variando únicamente en las herramientas con las que se realiza. Generalmente, se obtiene una muestra del medio a analizar, se le aplican una serie de procedimientos para resaltar el factor a medir, y se procesa en un medio visual por un técnico o un instrumento de laboratorio. Este último paso es crítico en casi todos los casos, puesto que la disponibilidad de los recursos implica que los resultados no suelen ser instantánteos. La tendencia en este aspecto es delegar en los equipos de laboratorio el procesamiento de las muestras, mientras que el técnico se encarga de obtener la muestra y supervisar el proceso. El proceso de análisis suele implicar la generación de imágenes digitalizadas que contienen la información a procesar. El análisis de estas imágenes normalmente resulta costoso (y en algunos casos inviable) de realizar únicamente por el personal científico sin hacer uso de los recursos informáticos. Normalmente, implica el uso de algún software que procese las imágenes y proporcione la información requerida. Este sofware suele ser bastante especializado y suele presentar un coste económico elevado.

En este trabajo proponemos realizar el procesamiento de las imágenes que llevan a cabo en los laboratorios en una plataforma remota haciendo uso de servicios web, con el fin de acelerar el proceso de análisis y reducir los tiempos de obtención de resultados, tal y como se expone en [1]. La principal aportación de los servicios web en este ámbito se refleja en la capacidad de tratamiento de las muestras en diferentes servidores no dependientes de los laboratorios, eliminando los retrasos producidos por la cola de procesamiento de resultados (puesto que los servidores pueden ser paralelos). Una ventaja importante es la descentralización de los recursos: se libera al laboratorio de equipos informáticos dedicados al análisis de las imágenes, reduciendo los costes de manteminiento y de espacio. Y otra aportación no menos importante es el aumento de los diferentes tipos de procesamientos que se pueden realizar, debido al abanico de servidores especializados en el análisis de imágenes.

La solución propuesta para procesar los datos de forma remota en este artículo se basa en la integración de un software de procesamiento de imágenes celulares, LLCECO [2], en una plataforma de servicios web. LLCECO es un software de clasificación y conteo de células desarrollado por el Grupo de Computación y Altas Prestaciones de la Universidad de La Laguna en colaboración con el Laboratorio de Biología Celular y Molecular de la Universidad de Castilla la Mancha. La plataforma de servicios web en la que ha sido integrado este software para ofertar el análisis de las imágenes celulares ha sido OpenCF [3]. LLCECO ha sido desarrollado desde cero para este proyecto y el diseño básico ha sido concebido como una herramienta para ser explotada desde la perspectiva de la computación de servicios.

El trabajo se ha estructurado como sigue: en la sección 2 se introduce el estado de las tecnologías actuales en el ámbito del artículo; la aplicación para contar células de forma autoguiada se expone en la sección 3; la sección 4 presenta OpenCF, el framework de servicios web utilizado en este proyecto; y la

integración de LLCECO en OpenCF se detalla en la sección 5. Un ejemplo de aplicación de este trabajo es presentado en la sección 6. Finalmente, la sección 7 muestra las líneas de interés en el ámbito del conteo de células y el trabajo futuro sobre la materia.

2. Trabajos relacionados

Tal y como se expone en [4], las plataformas basadas en servicios web están en alza. Grandes empresas y grupos de desarrollo de software como Rackspace, Microsoft, Google o Amazon apuestan por este tipo de plataformas como soluciones tecnológicas, y los laboratorios y grupos científicos no son ajenos a ellas [5]. El procesamiento de imágenes es una de las categorías de servicios web más demandadas por parte de los usuarios, debido al extenso ámbito de aplicaciones (desde reconocimiento faciales en imágenes en redes sociales hasta sofisticados algoritmos de procesamiento en imágenes astronómicas [6]), y en cualquiera de las plataformas anteriores encontramos una cantidad importante de servicios relacionados con este contexto (véase [7]).

En el ámbito biológico y médico encontramos herramientas de servicios web como Inbiomed [8], una plataforma de procesamiento de información biomédica que agrupa métodos, datos y análisis de imágenes en una única aplicación haciendo uso de servicios web, o caGrid [9]. Son plataformas de propósito general, con muchos servicios web que sustenten las necesidades de las empresas que están detrás de los proyectos. Un inconveniente de estas herramientas es que están orientadas a los centros de investigación que las desarrolla, por lo que su integración en otros ámbitos suele ser inviable de manera general.

Otro de los elementos importantes en el procesamiento de imágenes biomédicas es el software que se utiliza en los laboratorios y centros de investigación. Generalmente se trata de software propietario, casi siempre asociado al equipamiento (citrómetros, microscopios, etc). Existen alternativas más ligeras y portables, como CellC Cell Counter [10], Pixcavator [11] o GSA Image Analyser [12], más asequibles económicamente y más genéricas. Estas aplicaciones implementan los algoritmos básicos de procesamiento y conteo de imágenes, pero se restringen al análisis de una única imagen con un conjunto de parámetros aportados por el usuario. La principal desventaja frente a las plataformas de servicios web es la limitación a los equipos informáticos del laboratorio: si el número de imágenes a procesar haciendo uso de este tipo de software aumenta considerablemente, los procesos de análisis se pueden convertir en una tarea extensa en el tiempo y costosa en recursos de personal técnico.

3. LLCECO

LLCECO (La Laguna Cell Counter) es una aplicación de escritorio desarrollada para identificar y contar diferentes tipos de células (ver ejemplo en figura 1), en colaboración con el Laboratorio de Biología Celular y Molecular de la C.M. de la Universidad de Castilla la Mancha. Escrita en Python y C, esta

aplicación permite, dado un conjunto de imágenes de entrada, clasificarlas, obtener los parámetros de procesamiento óptimos de forma automática, identificar y contar los patrones (células), y generar documentos personalizados con los resultados. Consta de tres módulos, módulo de pre-procesamiento, procesamiento y post-procesamiento (ver figura 2), totalmente independientes entre sí.

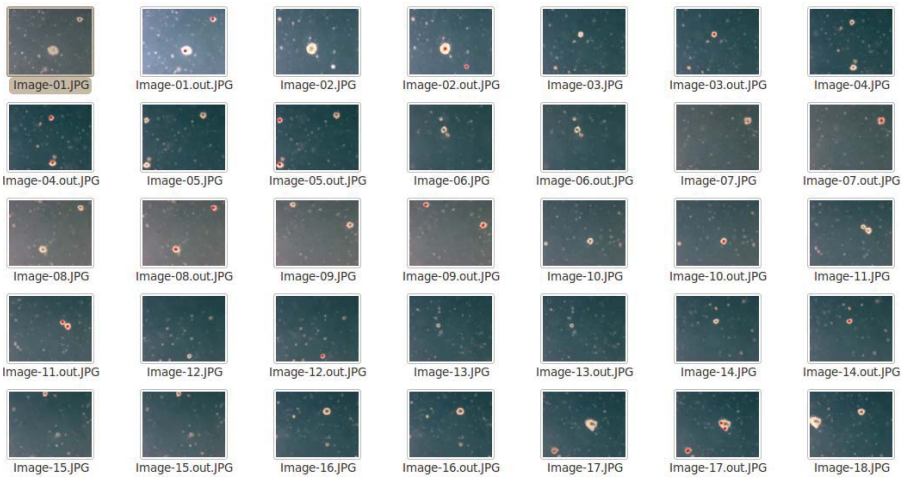


Figura 1. Directorio con las imágenes originales y las resultantes.

El módulo de pre-procesamiento es el encargado de recibir los argumentos de entrada, como son el fichero con las imágenes a procesar, parámetros para el procesado y para el post-procesado, gestionar la creación del árbol de directorios y descomprimir el archivo. Los directorios son generados por fecha de las muestras, y el resto de argumentos son enviados a los módulos correspondientes.

El módulo de procesamiento es el motor de la aplicación. Consta de 4 submódulos que se ejecutan secuencialmente:

- `Clasificación de la imagen`. Se halla la media de píxeles para ciertos rangos del esquema RGB, con lo que se obtiene información del fondo, colores de las células, etc.
- `Obtención de parámetros`. Una vez se tiene clasificada la imagen, se mapea con un conjunto de muestras previas con parámetros genéricos conocidos. Este conjunto está compuesto por cuatro tipos de imágenes celulares, y cada tipo ha sido obtenido procesando y clasificando conjuntos de entre 25 y 50 imágenes.
- `Conteo`. El algoritmo de conteo usado en LLCECO es la Transformada Circular de Hough [13]. Previamente se transforma la imagen al esquema de colores HSV y se aplica un filtro Gaussiano para eliminar el ruido y reducir la detección de círculos falsos.

- `Generación de resultados`. Como resultado se obtiene un documento con los datos de las imágenes (número de células, estadísticos, etc) y las imágenes originales superpuestas con una marca por cada una de las células identificadas.

El módulo de post-procesamiento es el más versátil y configurable, ya que depende de un conjunto de plantillas para generar los ficheros de datos. Consta de tres submódulos: la generación de gráficas, adaptación de las plantillas y agrupación de los resultados en un fichero comprimido. Las gráficas se generan a partir de los datos obtenidos en la fase de procesamiento, relacionan las muestras con el número de células obtenidas por imagen y por conjuntos. Para obtener ficheros de datos se utilizan plantillas excel que procesan los resultados (aplicación de métodos estadísticos, fórmulas, etc) dependiendo del tipo de análisis que se especifique en los parámetros de entrada. Y por último se genera un fichero comprimido, que será devuelvo al usuario, con las imágenes generadas, las gráficas y los distintos ficheros de datos.

LLCECO es una aplicación portable, ligera, sobre la que se están desarrollando nuevas funcionalidades. La facilidad para agregar nuevas plantillas ofrece un alto grado de libertad para adaptarla a las necesidades de cualquier laboratorio, y en su desarrollo se ha contemplado la posilidad de procesar las imágenes en computadoras paralelas para acelear la obtención de resultados.

Obsévese que el diseño modular de la herramienta admitiría la interacción de cada módulo, de forma independiente, con otros módulos/servicios que presentaran nuevas funcionalidades y que a su vez podrían ser ofertados como nuevos servicios. Esta facilidad admitiría la generalización de la plataforma de servicios hacia otros dispositivos de entrada/salida y hacia otros algoritmos de procesamiento de imágenes.

4. Plataforma de servicios web OpenCF

OpenCF es un framework computacional desarrollado por la Universidad de La Laguna. Es una plataforma ligera, portable, de fácil instalación, configuración y uso. La arquitectura software de OpenCF, mostrada en la figura 3, destaca por un diseño modular: módulo servidor y módulo cliente. Los módulos pueden ser extendidos independientemente e incluso reemplazados para proveer nuevas funcionalidades sin alterar el resto de componentes del sistema. OpenCF está orientado a máquinas paralelas y sus gestores de colas, por lo que el acceso, gestión de la ejecución y control de carga de la máquina está contemplado en la infraestructura.

El módulo cliente (front-end) y el módulo servidor (back-end) implementan las tres capas inferiores de la pila que describen los servicios Web: Descripción de Servicios, Mensajería XML y Transporte. El cuarto nivel, Descubrimiento de Servicios, no ha sido implementado por motivos de seguridad. La gestión del acceso de los usuarios al portal se realiza en el lado del cliente, mientras que la comunicación entre los módulos clientes y servidores se verifica mediante técnicas tradicionales de autenticación.

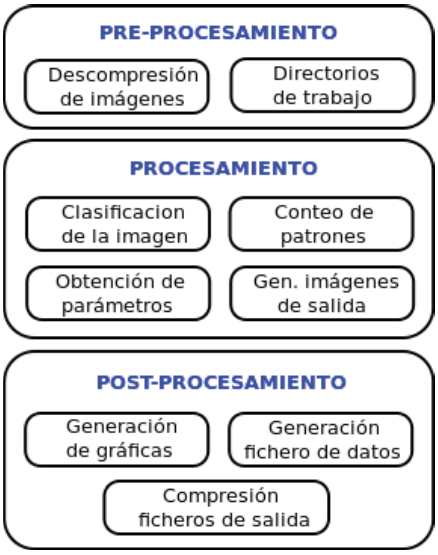


Figura 2. Módulos de LLCECO

El módulo servidor gestiona todas las cuestiones relacionadas con los trabajos, haciéndolos disponibles en el servicio y controlando su estado y ejecución. Cuando el servidor HTTP captura una nueva petición del cliente OpenCF, le asocia un hilo de ejecución en el que se crea una instancia independiente del módulo servidor.

El módulo cliente provee una interfaz para el usuario final y traduce las peticiones en consultas para el servidor. El servidor recibe las peticiones desde los clientes autenticados y las transforma en trabajos para el gestor de colas. Estos módulos, a su vez, están también modularizados. Los módulos Control de Acceso, Procesador de Peticiones y Recolector se pueden encontrar tanto en el lado del servidor como del cliente. El front-end también mantiene una base de datos para manejar la información generada por el sistema. El servidor incluye elementos para la generación de scripts y lanzamiento de trabajos en el sistema de colas.

El servidor OpenCF permite agregar servicios fácilmente a partir de procesos existentes, con independencia del software. Requiere de un fichero XML que describa la tarea (nombre, descripción y argumentos) para cada una de las rutinas que se deseen agregar como nuevos servicios web. Estos servicios serán ofertados automáticamente en los clientes OpenCF que tengan configurados el servidor, indicando el nombre del servicio, argumentos y una breve descripción de la rutina y el servidor que la oferta.

OpenCF implementa los servicios de monitorización y *scheduling* [14]. El cliente muestra los servicios comunes a distintos servidores agrupados en un único servicio, y basándose en la política de asignamiento implementada (sche-

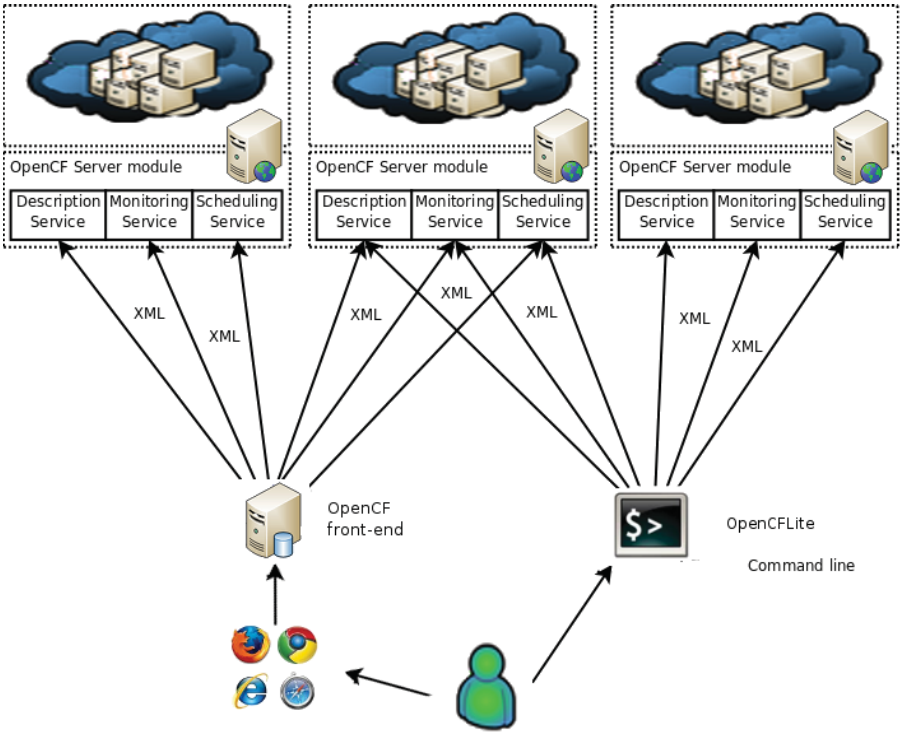


Figura 3. Arquitectura OpenCF

duling ordenado por el "potencial computacional normalizado"de los servidores) lanzará la ejecución al mejor servidor candidato.

5. Integración de LLCECO en OpenCF

La infraestructura que hemos desarrollado auna la plataforma de servicios web OpenCF y la aplicaciones LLCECO en un único concepto, con las consiguientes ventajas que esto implica. Al integrar LLCECO en OpenCF conseguimos que se puedan procesar distintos tipos de imágenes de forma remota, simple y eficientemente. Se justifica la utilización de OpenCF frente a otras herramientas semejantes (como OpenNebula o Eucalyptus [15]) debido a factores como la facilidad para integrar módulos externos (LLCECO) y la libertad para trabajar en arquitecturas paralelas. La integración se ha realizado en dos pasos: desarrollo de una capa de encapsulación de la aplicación y descripción del servicio.

La comunicación entre OpenCF y LLCECO se produce a través de una capa de encapsulación de la aplicación, que traduce las peticiones del servidor a comandos de la aplicación, tanto para el paso de parámetros como para la recu-

peración de los resultados. Esta capa se basa en un script de Python, encargado de leer la entrada desde línea de comandos y comunicarse con la aplicación LLCECO. El servidor web recoge los parámetros a través de su interfaz web (ver figura 4) y realiza la llamada a LLCECO mediante el script de encapsulamiento, pasando los distintos parámetros como argumentos. Una vez finaliza la ejecución de LLCECO, el script comprime los resultados en un fichero que será copiado al índice de resultados del servicio web, asociándolo al ID de la tarea, permitiendo que el usuario recupere el fichero a través de la interfaz del cliente OpenCF (también le será avisado mediante un correo una vez que el servidor OpenCF reciba los resultados de la ejecución).

Figura 4. Servicio web de LLCECO en OpenCF

La descripción del servicio LLCECO en OpenCF se realiza mediante un fichero XML donde se especifican el nombre de servicio, una breve descripción y el conjunto de parámetros de entrada y salida (nombre, tipo de argumento y descripción, ver Listing 1.1).

Se especifica el nombre del servicio (*cell_count*), los argumentos de entrada (*images_zip* y *factor*) y el argumento de salida (*output_zip*). Estos nombres han de ser los mismos en cada uno de los servidores en los que se integre LLCECO, de manera que se pueda realizar una planificación sobre los servicios en los clientes web de OpenCF.

Una vez desarrollados el script de encapsulamiento y el documento XML que describe el servicio, se instala la aplicación LLCECO en el servidor y se configura el script con el directorio raíz, número de hilos a ejecutar y directorio de resultados del servidor OpenCF. Se copia el documento XML en el directorio del servidor correspondiente y se añade el nuevo servicio web al servidor OpenCF mediante el binario *add_job.sh* pasando como argumentos el nombre del fichero de descripción del servicio.

Listing 1.1. imgs/code.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="job.xsl"?>
<job xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNameSpaceSchemaLocation="job.xsd">
   <name>Cell count</name>
   <service_name>cell_count</service_name>
   <wclient_id>opencf-dev_pcg_ull_es</wclient_id>
   <user_id>test</user_id>
   <binary>bin/cell_process.py</binary>
   <description>
      Example of a matrix invert operation in R
   </description>
   <argument type="base64Binary">
      <name>images_zip</name>
      <sdesc>Zip file with cells image</sdesc>
      <ldesc>
         Zip containing a directory with images of cells
      </ldesc>
      <fname_in_server>images.zip</fname_in_server>
   </argument>
   <argument type="integer">
      <name>factor</name>
      <sdesc>Factor to multiply cell numbers</sdesc>
   </argument>
   <output_file>
      <name>output_zip</name>
      <description>
        All the input and results files of the execution
        of the service.
      </description>
   </output_file>
</job>
```

6. Caso de uso

Haciendo uso de la plataforma hemos procesado un conjunto de muestras pertenecientes al Labotario de Biología Celular y Molecular de C.M. de la Universidad de Castilla la Mancha. La información está contenida en dos directorios que contienen las imágenes relativas a dos cultivos con 25 y 40 imágenes respectivamente. Se registra una única petición a la plataforma en la que se incluyen los dos directorios en un archivo comprimido (aproximadamente unos 6MB, con tiempos de subida de varios segundos para la red de investigación, y menor de un minuto para una red doméstica). El sistema devuelve un archivo comprimido en el que se encuentran los directorios de imágenes procesados y la hoja de cálculo con los datos obtenidos y las gráficas generadas. El procesamiento de las imágenes y la obtención de los resultados tiene una duración de pocos segundos,

por lo que el cuello de botella de la aplicación está en el envío y recepción de los datos (en caso de aumentar el tamaño de archivo, se dividiría en varios para optimizar el proceso).

Si el número de peticiones es elevado, haciendo uso de un clúster de cómputo se mantendrían los tiempos de ejecución en estos rangos y el servicio podría escalar a un número elevado de peticiones. La figura 1 muestra la salida correspondiente a uno de los directorios. La figura 5 muestra la hoja de cálculo resultante para este experimento en el que se ha realizado un análisis estadístico tipo ANOVA.

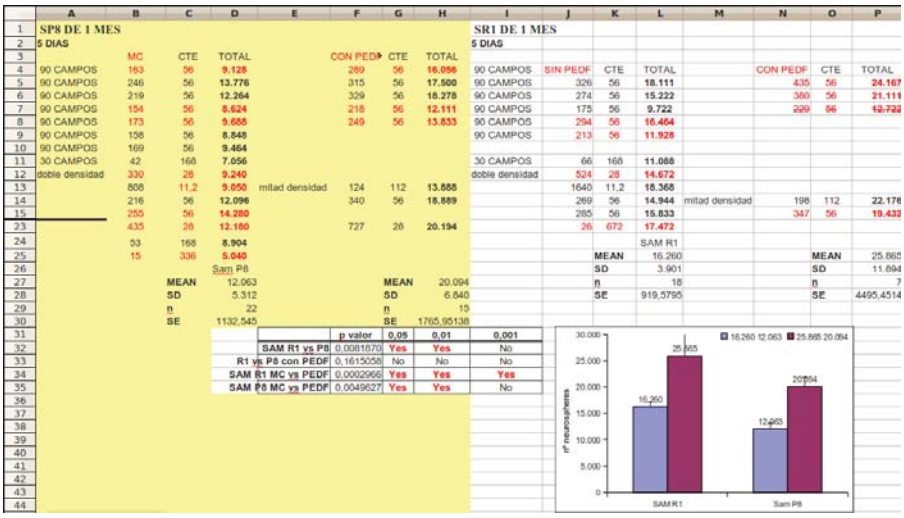


Figura 5. Hoja de cálculo resultante con los datos, análisis estadístico y gráficas.

7. Conclusión y trabajo futuro

En este artículo se presenta una plataforma de computación genérica basada en servicios web para problemas de conteo de células en imagénes biomédicas. Esta herramienta, compuesta por la integración de una plataforma de servicios web (OpenCF) y una aplicación para procesar y contar células (LLCECO), permite ejecutar desde una interfaz web el procesamiento y recuento de células en las imágenes de varios directorios (comprimidos en un fichero, conformando el argumento de entrada al proceso), obteniendo como resultado una serie de ficheros con los datos del análisis y las imágenes de salida de la aplicación. Con esta plataforma pretendemos mejorar el ritmo de trabajo en los laboratorios e investigaciones que precisen de analizar y contar células en sus experimentos reduciendo los costes y los tiempos de obtención de resultados.

Esta plataforma se diferencia de las existentes hasta ahora por la mayor flexibilidad y portabilidad, generalidad de los servicios y capacidad de procesamiento. Haciendo uso de servicios web conseguimos utilizar una nube de servidores que procesan las imágenes de forma transparente al usuario, con una capacidad de cómputo mayor que los equipos existentes en los laboratorios.

Las líneas de desarrollo actuales sobre la plataforma se centran en el aprendizaje autoguiado de la clasificación y obtención de parámetros óptimos para el análisis de las imágenes, de forma que la ejecución de las tareas sea totalmente autónoma por parte del servidor; y en los casos de uso de las imágenes actuales, realizando experimentos con imágenes topográficas, de cultivos o espermiogramas.

Como mejoras se propone un cliente ligero multiplataforma para los equipos de laboratorio encargados de la obtención de la imagen, de forma que se realice el proceso de ejecutar la tarea con los parámetros correspondientes en la plataforma de forma automática, sin intervención del usuario.

8. Agradecimientos

Este trabajo ha sido parcialmente financiado por el EC F(EDER) y la MICINN española (Plan Nacional de I+D+I, TIN2008-06570-C04-03 y TIN2011-24598).

Referencias

1. Solomonides T, M.R.e.B.V.: From grid to HealthGrid. In: Technology and Informatics, Amsterdam, Netherlands (September 2005)
2. Castillo, J.: Llceco: La laguna cell counter (2012) http://opencf.pcg.ull.es/redmine/projects/opencfcc.
3. Santos A, A.F., V, B.: Lightweight web services for high performace computing. European Conference on Software Architecture ECSA2007 **4758** (2007)
4. Darrow, B.: Amazon is no.1 who's next in cloud computing (2012)
5. Evangelinos C, H.C.: Cloud computing for parallel scientific hpc applications: Feasibility of running coupled atmosphere-ocean climate models on amazons ec2. First workshop on Cloud Computing and its Applications, Chicago, USA (2008)
6. Almeida, F., Blanco, V., Delgado, C., de Sande, F., Santos, A.: Idewep: Web service for astronomical parallel image deconvolution. J. Netw. Comput. Appl. **32**(1) (January 2009) 293–313
7. Fronckowiak, J.: Processing images with amazon web services (2008)
8. D.Perez, J. Crespo, A.A.: Biomedical image processing integration through inbiomed: A web services-based platform. Biological and Medical Data Analysis (2005) 34–43
9. Zhang, J., Madduri, R., Tan, W., Deichl, K., Alexander, J., Foster, I.: Toward semantics empowered biomedical web services. In: Web Services (ICWS), 2011 IEEE International Conference on. (july 2011) 371 –378
10. Jyrki Selinummi, Jenni Seppälä, O.Y.H., Puhakka, J.A.: Software for quantification of labeled bacteria from digital microscope images by automated image analysis. BioTechniques **39**(6) (2005) 859–863

11. Saveliev, P.: Pixcavator (2011) http://inperc.com/wiki/index.php?title=Cell counting.
12. GSA: Gsa image analyser (2012) "http://image.analyser.gsa-online.de/".
13. OpenCV: Hough circle transform (2012) http://opencv.itseez.com/doc/tutorials /imgproc/imgtrans/hough_circle/hough_circle.html.
14. Santos A, Almeida F, B.V., J.C, C.: Web services based scheduling in opencf. Journal of SuperComputing (2010) 88–114
15. Sempolinski, P., Thain, D.: A comparison and critique of eucalyptus, opennebula and nimbus. Critique **November** (2010) 417–426

Definición y Aplicación de un proceso de Modernización y Evolución al Sistema de Gestión de Nombres de Dominios ".es"

Jorge Moratalla Collado[1], Esperanza Marcos[2]

[1]Entidad Pública Empresarial Red.es
(Ministerio de Industria Turismo y Comercio - Gobierno de España) Madrid-ESPAÑA
jorge.moratalla@red.es

[2]Grupo de Investigación KYBELE
Universidad Rey Juan Carlos. Móstoles (Madrid-ESPAÑA)
esperanza.marcos@urjc.es

1 Antecedentes

En los últimos años, la necesidad de evolución y modernización de los sistemas heredados se ha convertido en uno de los principales problemas a los que se enfrentan las organizaciones, ya sea para incorporar nuevas funcionalidades o para adaptarse a diferentes plataformas tecnológicas. En la Administración Pública este aspecto es aún más crítico. La optimización de los recursos se convierte en un factor clave a la hora de realizar los procesos de evolución y modernización. Tal es el caso de Red.es, entidad pública vinculada al Ministerio de Industria, Energía y Turismo, cuya estrategia para el impulso de la sociedad de la información hace hincapié en la necesidad de adaptar sus aplicaciones y sistemas de información a la nueva legislación relativa a la interoperabilidad y la administración electrónica, entre otros.

La estrategia de la Entidad para conseguir este objetivo, en la actual coyuntura económica, está marcada por la reducción de tiempos y costes, la externalización y fragmentación del desarrollo, y la maximización del retorno sobre la inversión (*ROI*). En este sentido, la optimización de los recursos se convierte en un factor clave a la hora de abordar este proceso [7].

Todos estos factores han sido abordados previamente de forma exitosa, en proyectos similares de otras organizaciones, utilizando el paradigma de orientación a servicios (*SOC, Service Oriented Computing*) [13][3][8][11]. Para apoyar el proceso de evolución y modernización, resulta esencial el uso de *gap analysis* [14] para proporcionar una manera de identificar la forma en la que los servicios software disponibles pueden ser ensamblados dentro de otros nuevos [6][9].

Una de las características distintivas del *gap analysis* es que funciona en un alto nivel de abstracción. Sin embargo, el proceso de evolución y modernización comienza normalmente a un nivel más tecnológico. Como consecuencia, el uso exclusivo del *gap analysis* con este propósito no es suficiente. En ese contexto, el *Object Management Group* (OMG) ha propuesto ADM (*Architecture Driven Modernization*) [2] como un enfoque para la evolución y modernización de los sistemas de información. ADM se basa en la definición y la utilización de un conjunto de modelos

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 77-80, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

y la aplicación de reglas de transformación entre ellos con objeto de: a) obtener modelos de alto nivel de abstracción a partir del análisis de las aplicaciones TI existentes, y b) realizar un análisis *top-down* de las nuevas aplicaciones de TI. Los principales metamodelos de ADM son el *Knowledge Discovery Metamodel* (*KDM*) [3], que representa los aspectos de la solución tecnológica y la estructura lógica del sistema, y *Semantic for Business Vocabulary and Rules (SBVR*) [10], cuyo objetivo permite definir la semántica y las reglas de negocio de la empresa.

El propósito de este trabajo es, por tanto, describir el proceso para la evolución y modernización definido, al que hemos denominado PREMISA, y su aplicación a Red.es. Los pilares de este proceso son la evolución y modernización dirigida por modelos basada en ADM, y el análisis del gap entre los modelos de alto nivel.

2 Descripción del problema a resolver

Para desarrollar la labor que la Entidad Pública Empresarial Red.es tiene encomendada (en lo relativo a la asignación y gestión de nombres de dominio), se dispone de un Sistema Informático, que se concibió inicialmente en un contexto en el cual el parque de los ".es" era reducido. Sin embargo este número ha ido creciendo de forma exponencial hasta llegar actualmente a gestionar más de 1'5 millones de ".es".

Para adaptarse a esta situación, el sistema ha ido evolucionando a demanda, mediante la aplicación de sucesivos parches, resultando en un sistema heredado difícil de evolucionar y modernizar, y poco documentado. Para dar respuesta a la nuevas reglas de negocio [9], se hace necesario aplicar un proceso de modernización y evolución. Este proceso será aplicado al módulo correspondiente a la transmisión de dominios (disponible en la URL http://www.dominios.es).

3 La solución propuesta: Aplicación a Red.es

El proceso que hemos definido en Red.es se muestra en la figura 1. Se basa en la transformación denominada "en herradura" [14], para conseguir el sistema objetivo partiendo del sistema actual mediante la ejecución de una serie de pasos.

El primer paso de nuestra propuesta consiste en representar el modelo lógico del sistema actual (*as-is*), utilizando la notación del metamodelo KDM, mediante la utilización de la herramienta MoDISCO [12] para la extracción semiautomática del modelo KDM. El siguiente paso consiste en refinar el modelo lógico (KDM) obtenido en el paso anterior, debido a que MoDISCO sólo permite el descubrimiento automático para las capas más cercanas a la tecnología.

Una vez refinado el modelo lógico, el siguiente paso consiste en obtener la capa de abstracción del modelo KDM. Mediante la aplicación de reglas de transformación al modelo lógico refinado, se obtendrá una representación de un mayor nivel de abstracción, cercana al nivel de negocio.

El siguiente paso consiste en obtener el modelo SBVR, a partir de las reglas de transformación definidas, que permitirá derivar las reglas de negocio (*Business Rules*) inherentes al sistema. Una vez obtenido el modelo de negocio de sistema actual, los

analistas de negocio modelan la solución objetivo a nivel de negocio utilizando SBVR, para que en el siguiente paso se pueda aplicar el gap analysis entre ambos modelos as-is y to-be. Este paso se realizará en tres etapas: representar los modelos de negocio en una notación formal, aplicar los operadores para identificar las similitudes y diferencias entre ambos modelos y obtener las operaciones a aplicar para transformar uno en otro.

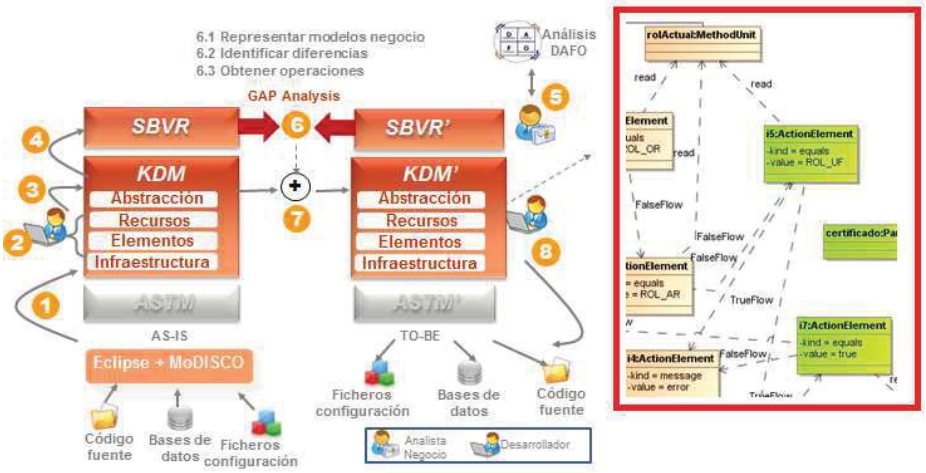


Figura1. Esquema global de PREMISA

Una vez definidas las nuevas reglas de negocio, el siguiente paso consiste en aplicar las operaciones obtenidas en el gap analysis al modelo KDM, para obtener el correspondiente a la solución objetivo, aplicando operaciones equivalentes y, por último actualizar el código fuente de la solución final, utilizando las mismas reglas de transformación en sentido inverso.

Tras dos iteraciones, el proceso de modernización se ha aplicado de forma exitosa en Red.es, para modernizar y evolucionar el sistema de gestión de nombres de dominios ".es". Las nuevas reglas de negocio añadidas permiten a un usuario final realizar la transmisión de sus nombres de dominio, siempre que sea el titular del mismo y se autentique de forma fuerte mediante un certificado digital.

La figura 1 muestra una parte del modelo lógico KDM de la solución to-be, obtenido mediante la sucesiva aplicación de las reglas de transformación correspondientes a los pasos definidos en el proceso. En verde se muestran los nuevos elementos añadidos de acuerdo a las nuevas reglas.

4 Conclusiones y lecciones aprendidas

Las principales conclusiones de la definición y aplicación del proceso podemos afirmar que se ha realizado una aplicación satisfactoria del proceso, al caso real de Red.es, además de establecer puntos de mejora para futuros trabajos. La aplicación se encuentra actualmente desplegada en un entorno productivo.

Otro punto importante es la institucionalización y estandarización del proceso, como semilla para los proyectos futuros de evolución y modernización en la Entidad, como base para la mejora continua (el control de este proceso y la estandarización de las tareas, permitirá utilizar los mecanismos necesarios para aplicar métricas).

Como lecciones aprendidas podemos mencionar que, aunque se ha definido un conjunto de reglas de transformación para iterar de un modelo a otro, este proceso es muy propenso a errores si no se realiza automáticamente. Por lo tanto, el desarrollo de un conjunto de herramientas de apoyo sería de gran utilidad.

Del mismo modo, el enfoque metodológico ofrecido por PREMISA no está orientado a la metodología de Red.es, por lo que sería necesario adaptarla.

Otro de los problemas encontrados ha sido la falta de especificaciones técnicas a la hora de modelar la solución objetivo. Estas especificaciones (tipo de datos, invocaciones a servicios, etc.) han debido ser consensuadas entre los desarrolladores y los analistas de negocio, una vez obtenido el modelo de negocio del to-be.

Referencias

[1] Advanced Signature Framework (ASF). https://asf.demo.red.es/asf_demo/

[2] Architecture Driven Modernization specification of the OMG, http://adm.omg.org/

[3] Arsanjani, A.; Ghosh, S.; Allam, A.; Abdollah, T.; Ganapathy, S.; Holley, K. SOMA: a method for developing service-oriented solutions, IBM Systems Journal 47 (3) (2008) (published online August 6, 2008).

[4] BOE 129 de 31 de mayo de 2005. Orden Ministerial ITC/1542/2005, de 19 de mayo, Plan Nacional de Nombres de dominio de Internet bajo ".es"

[5] BOE número 129 de 31/05/2005. Orden ITC/1542/2005, de 19 de mayo (Apartado Segundo, Disposición Primera)

[6] Bolstorff, P., Rosenbaum, R.: Supply Chain Excellence: A Handbook for Dramatic Improvement Using the Scor Model. Ed. AMACOM, 2nd edition (2007).

[7] Clavreul, M.; Barais, O.; Jezequel, J.; "Integrating Legacy Systems with MDE". ACM Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, May 2010

[8] Erl, T. (2005). Service-Oriented Architecture: Concepts, Technology, and Design. Upper Saddle River: Prentice Hall PTR.

[9] Instrucción del Director General de red.es, de 2 de enero de 2010

[10] Jeston, J.; Nelis, J. "Business Process Management: Practical Guidelines to Sucessful Implementations" Ed. Butterworth-Heinemann, 3rd edition (2006)

[11] Krafzig, D.; Banke K., Slama D. Enterprise SOA Service Oriented Architecture Best Practices. Upper Saddle River: Prentice Hall PTR. 2004

[12] MoDISCO user guide, available at http://help.eclipse.org/helios/index.jsp?nav=/38

[13] Papazoglou, M.; Traverso, P.; Dustdar, S.; Leymann, F.; "Service-Oriented Computing: A research roadmap". INTERNATIONAL JOURNAL OF COOPERATIVE INFORMATION SYSTEMS, ISSN 0218-8430, 06/2008, Vol. 17, Nº 2, pp. 223 – 255

[14] Ulrich, W. "A Model - driven Approach to Modernizing Existing Systems" Architecture Driven Modernization Workshop, Chicago, Marzo 2004.

[15] Nguyen, D.K., van den Heuvel, W.J., Papazoglou, M., de Castro, V., Marcos, E. GAMBUSE: A Gap Analysis Methodology for Engineering SOA-Based Applica-tions. Conceptual Modelling: Foundations and Applications Essays in Honour of John Mylopoulos, LNCS 5600, pp.293-318, Springer Verlag, (2009).

Clasificación de los Servicios Web de Negocio Corporativos basada en la Funcionalidad Horizontal de las Organizaciones

Miguel A. González Serrano[1], Diana Pérez-Marín[2], Miren Idoia Alarcón[1]

[1]Universidad Autónoma de Madrid, Spain
[2]Universidad Rey Juan Carlos, Madrid, Spain
miguela.gonzalez@estudiante.uam.es, diana.perez@urjc.es, idoia.alarcon@uam.es

Abstract. Basándonos en la experiencia adquirida en la realización de tres proyectos que plantean arquitecturas empresariales, se han definido 14 tipos de servicios de negocio que deben ser considerados en las etapas de análisis y diseño de Arquitecturas Orientadas a Servicios. El uso de los servicios identificados en este artículo podría conllevar una mejora efectiva y temporal de alrededor de un 40% en tiempo y esfuerzo de las fases de análisis y diseño de desarrollos informáticos en grandes entidades corporativas.

Palabras clave: SOA, Servicios Web, Clasificación de Servicios.

1 Introducción

Durante los últimos años, la Arquitecturas Orientadas a Servicios (SOA) y la Computación Orienta a Servicios (SOC) se han convertido en áreas de investigación [1-3] que proponen nuevos paradigmas en el tratamiento de la complejidad de las aplicaciones distribuidas. En este contexto, las Arquitecturas Orientadas a Servicios (SOA) son un medio de captura de principios, guías y técnicas que proporcionan un modelo para el desarrollo de estas aplicaciones dentro de una arquitectura. En el desarrollo de 3 proyectos en grandes organizaciones de entre 2000 y 5000 empleados se ha detectado que se pueden mejorar las fases de análisis y diseño de los proyectos que plantean la implantación de arquitecturas empresariales gracias a la aplicación de Frameworks comunes [4-7]. Los proyectos analizados sirven de base como evidencia del buen funcionamiento de la propuesta de este estudio y responden a diferentes casos de negocio tales como la implantación de una arquitectura de movilidad, la reingeniería de la arquitectura de gestión de errores técnicos y de negocio en una organización de *"core"* de negocio bancario y la implantación de una arquitectura de procesos de negocio en una organización cuyo negocio es la gestión de contenidos.

Actualmente, las metodologías y tipologías existentes son generalistas [8-14], dejando de lado los conceptos más puramente prácticos y cercanos al negocio y las características tecnológicas que rigen las organizaciones.

El propósito de este artículo es, por una parte, proponer un elemento de mejora para el análisis y diseño de arquitecturas SOA, dentro de un ciclo de vida tradicional, por medio de la caracterización de nuevos tipos de servicios, por otra, que esta taxonomía sirva para evitar las limitaciones típicas tales como una gran dependencia

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 81-87, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

de sistemas, la creación de interfaces altamente dependientes de la tecnología, existencia de un gran gap funcional entre las especificaciones de negocio y la implementación tecnológica y la falta de uso de estándares.
Los tipos de servicio de negocio actualmente identificados en el estado del arte se estudian críticamente buscando sus limitaciones en relación a los ciclos de vida SOA y proponiendo nuevos tipos de servicios que permitan paliar las deficiencias encontradas en el análisis de los actuales ciclos de vida del software de proyectos SOA (centrándose en las fases de análisis y diseño).

El artículo está organizado en tres apartados: en el apartado 2 se revisan los enfoques más relevantes sobre clasificación de servicios y sus metodologías en arquitecturas SOA. En el apartado 3 se explica la contribución de este trabajo con respecto al estado del arte. Por último el apartado 4 termina con la discusión de los conceptos propuestos y algunas conclusiones.

2 Estado del arte

Las metodologías de SOA difieren unas de otras en función del momento en la historia en la que han sido planteadas, unidas a paradigmas tecnológicos que regían el momento y sobre todo a necesidades de negocio que mandaban sobre la tecnología. Algunos autores se basan en modelos teóricos [9, 11, 12] y otros lo hacen sobre estudios realizados en proyectos reales [10, 15-16]. En la Tabla 1 se resumen las principales propuestas de taxonomías de servicios incluidas en las metodologías contemporáneas.

Tabla 1. Principales aportaciones metodológicas a SOA

Autor, año, referencia	Alcance
Zachman (1987) [4]	Framework de Arquitecturas Empresariales
Arsanjani (2004) [11]	SOMA: Requisitos funcionales y no funcionales. Arquitec. de capas.
Zimmermann, et al (2004) [13]	EADI - Uso de UML en SOA
Zimmermann, et al (2004) [14]	SOAD (y su relación con BPM, EA y OOAD
Huhns, Singh (2005) [12]	Impacto en las aplicaciones inter-empresa e intra-empresa.
Erl (2005, 2007) [9, 18]	Conceptos en el análisis y diseño de SOA. Principios y patrones SOA
Doddavula (2005) [1]	Framework de desarrollo SOA
Jones(2005)[10]	Ciclo de cuatro pasos (Qué, quién, por qué y cómo). Servic. Virtuales
Papazoglou (2007) [2,3]	Aspectos técnicos y de más bajo nivel de arquitecturas SOA.
López, et al(2007) [20]	Ciclos de vida SOA en procesos de negocio y su orquestación.
Ramollari, et al (2007) [21]	Metodologías SOA y sus ciclos de vida.
Caicedo, et al (2008) [15]	Adopción de SOA. Ciclos metodológicos clásicos y sus fases.
Delgado, et al.(2010)[7]	MINERVA System.
Joachim, et al (2010) [8]	Factores en la adopción de SOA
NEXOF-RA [22]	Framework de Planteamiento de arquitecturas orientadas a servicios.
S-Cube [23]	S-Cube
SOD-M (2007)[24]	Aproximación MDA para el desarrollo orientado a servicios.

3 Clasificación de Servicios de Negocio en Arquitecturas empresariales

Los autores Erl [9, 17-18], Doddavula [3] y Papazoglou [4-5] centran el diseño de servicios a partir de tipologías generalistas en las que encontramos servicios que tratan de cubrir muchos aspectos desde un punto der vista particularizado en la tecnología (Web Services). Estos enfoques dejan escapar numerosos aspectos funcionales y tecnológicos que deben reflejarse en una arquitectura empresarial basada en SOA a día de hoy, por ello este artículo propone que la base de una metodología SOA necesita de conceptos más cercanos al negocio y sus sistemas empresariales.

Por ello realizamos como una de las principales aportaciones un listado y descripción de los tipos de servicios que enriquecerían el enfoque de Erl [9, 17-18] y autores como Arsanjani [11, 19] (estrategia SOA no empieza por un análisis botom-up), enfoque es más típico de una solución de Web Services, pero no necesariamente, una solución SOA está compuesta en su totalidad en Web Services [9, 19], se trata de algo mucho más complejo y estratégico para el negocio de una organización. Es por eso que nuestro enfoque se centra en servicios y horizontales que dan soporte a los servicios de negocio global, sin centrarse en una de tecnología o en categorizaciones de servicios demasiado generalistas.

Por otro lado, desde una perspectiva más global teniendo en cuenta las metodologías SOA como un todo, autores como Zimmermann [13, 14] o Huhns [12] afirman respectivamente que el análisis y diseño de servicios debe ser planteado como el refinamiento de técnicas existentes en diferentes paradigmas tecnológicos y sus tecnologías, tales como BPM, OOAD y EA. Sin embargo, estos paradigmas tecnológicos no cubren todos los aspectos requeridos por SOA y es necesaria una ampliación de estas metodologías. La categorización propuesta ayuda a conseguir este objetivo. A continuación se describen los servicios que son propuestos por este análisis, Figura 1, esta tipología de servicios de negocio y IT, surge como resultado del estudio del estado de arte de este artículo así como del resultado de analizar e implantar diversos proyectos en grandes entidades por parte de los autores de este articulo.

Figura 1. Tipología de servicios propuesta

Los principales objetivos de SOA se pueden lograr desarrollando un plan estratégico que permita alinear los recursos de IT con los objetivos de negocio [6]. Una solución completa que plantee una Arquitectura Empresarial [4, 5] debe abordar la semántica de la terminología y también los niveles de abstracción del negocio [4, 5], por ello este articulo plantea que los recursos IT, sistemas y necesidades tecnológicas que son comunes y se han detectado en todas las organizaciones analizadas que se plantean instaurar una arquitectura orientada a servicios de negocio:

- **Seguridad**: Un modelo de seguridad integrado dentro de una aplicación, puede

ya no ser apropiado cuando las funciones de la aplicación están expuestas como servicios. Actualmente están surgiendo nuevos modelos de seguridad para SOA [15], queda patente pues la necesidad de la creación de servicios de arquitectura de seguridad.

- **Gestión de contenidos (CMS)**: La industria experimenta numerosos cambios que hacen crecer la necesidad de la compartición de contenidos empresariales entre diferentes organizaciones, esto sumado a que los actuales CMS (Content Management Systems). Por ello se están desarrollando estándares como CMIS de OASSIS. Es por ello que un servicio de contenidos basado en CMIS deba ser considerado para una plataforma SOA.
- **Gestión documental (ECM)**: La importancia de estas tecnologías se hace patente en todos los informes tecnológicos que Forrester y Gartner [6, 19] publican anualmente. Es por ello que apostamos por una arquitectura con un servicio corporativo de gestión de documentos que planteado como un servicio de negocio sea accesible por todas las aplicaciones corporativas.
- **Gestión de relaciones con clientes (CRM)**: La gestión de relaciones con clientes es otro de los dominios funcionales que todas las organizaciones analizadas comparten. Es importante considerar un bloque funcional dentro de nuestra propuesta de arquitectura que solucione este tipo de negocio .
- **Presentación**: Tradicionalmente, el "Front-End" se ha incluido en la capa de negocio. Con el paradigma OOAD esta separación se hace patente con el concepto de capas, aportando la capa de presentación como un elemento separado del negocio de la aplicación. Proponemos que una organización debe estar dotada de servicios de presentación que permitan presentar contenidos en diferentes formatos y estilos para distintas tecnologías.
- **Movilidad**: Con los SmartPhones, las organizaciones registran requisitos de movilidad. Por todo ello planteamos muy necesario que una arquitectura corporativa basada en SOA tenga entre sus tipos de servicios algunos dedicados a la producción de información destinada a las aplicaciones de movilidad.
- **Integración de aplicaciones o datos**: Este tipo de servicios queda patente con la aparición de la necesidad de un Modelo Común de Datos horizontal a toda la organización. A este modelo común se accede en situaciones que se desea traducir las entidades de negocio para proveer a la arquitectura empresarial de un único sistema de datos para todas las aplicaciones corporativas.
- **Lógica de adaptación o multificación**: Tradicionalmente las aplicaciones se dividen en capa de presentación, lógica de negocio y datos [9]. Sin embargo, dada la creciente necesidad de adaptar una misma lógica a varios tipos de negocio segmentados por canales negocio diferentes, se hace necesario definir la Lógica de Adaptación que permita adaptar servicios al negocio.
- **Internacionalización**: Dentro de la lógica de adaptación de un canal se consideran también el subtipo de adaptación al idioma y aspectos relativos a la internacionalización del negocio y presentación de una aplicación.
- **Reglas de negocio**: Las organizaciones se rigen gracias a reglas de negocio, explícitas o tácitas. Es de vital importancia la creación de un tipo de servicio que sirva las reglas de negocio a las aplicaciones de una organización, aislando el mantenimiento de las mismas y minimizando el impacto en los cambios sobre cualquiera de ellas.

- **Servicios de Administración**: Es necesaria las funciones de Administración de esos servicios. Se distinguen tres subtipos, **Gobierno**, **Notificación** y **Configuración**.
- **Auditoria:** El principal requisito de la auditoría es que la explotación de la información operacional de aplicaciones corporativas sea online (tiempo cercano a real). Se distinguen tres subtipos de esos servicios, Auditoria de aplicaciones, Auditoria de Negocio, Auditoria técnica de arquitectura.
- **Servicios de Back-Office**: Este tipo de servicios en una arquitectura SOA se encargaría de proveer funcionalidad puramente de negocio a las diversas piezas de la arquitectura que lo necesitasen.
- **Soporte a Procesos de Negocio:** Se consideran los servicios de soporte a negocio para cubrir una amplia gama de funcionalidad, la gestión de procesos de negocio (BPM). Las principales aportaciones que ofrecen estos servicios se encuentran el soporte a configuración, ejecución de eventos de negocio, la automatización de la ejecución de procesos, etc.

4 Discusión

Se ha podido comprobar cómo utilizar estos servicios en las fases de análisis y diseño de arquitecturas SOA conlleva una mejora efectiva y temporal de alrededor de un **40% en tiempo y esfuerzo**. Otros beneficios identificados son los siguientes:

- El servicio de seguridad permite garantizar que la seguridad será global y horizontal a la organización.
- Los servicios de negocio de CMS, ECM y CRM se consigue un acercamiento al negocio (reutilización y combinación funcional).
- Los servicios de presentación consiguen el formateo de información no esté ligado a un sistema o tecnología concreta.
- Los servicios de movilidad integran esta nueva arquitectura con los sistemas existentes y tradicionales de la organización.
- Los servicios de integración de aplicaciones y datos posibilitan la definición de modelos reutilizables por la organización favoreciendo el intercambio.
- La lógica de adaptación o “multificación” suponen una personalización dentro de marco internacional que reduce el coste de desarrollo.
- Con los servicios de reglas de negocio se limita el impacto en los sistemas ante una nueva funcionalidades.
- Los servicios de gobierno o notificación y auditoria de aplicaciones ayudan a gobernar y monitorizar las arquitecturas empresariales.
- Los servicios de soporte a negocio son cruciales para llenar una necesidad que entidades corporativas demandan hoy en día, la gestión del negocio desde un punto de vista de procesos.

El uso de estos servicios deberá tenerse en cuenta en relación a un marco metodológico englobado en las fases de análisis y diseño de una arquitectura empresarial. La aplicación de cada uno de los tipos de servicios está sujeta a la necesidad directa de negocio o técnica del tipo en cuestión.

Finalmente y como trabajo futuro se plantea realizar un marco de referencia completo dentro de una metodología de planteamiento de una arquitectura SOA donde se incluyan modelos automáticos tales como el plateado por [24] en lo que supone una aproximación MDA para el desarrollo orientado a servicios de sistemas de información

Referencias

1. Doddavula, S.K.: Designing an Enterprise Application Framework for Service-Oriented Architecture, Java.Net, 2005.
2. Papazoglou, M.P., van den Heuvel, W.J.: Service-oriented design and development methodology, International Journal of Web Engineering and Technology (IJWET), 2006.
3. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-Oriented Computing: State of the Art and Research Challenges, IEEE Computer Society, 2007.
4. Zachman, J.A. A framework for information systems architecture, Ibm Systems Journal, Vol. 26. No. 3, 1987.
5. Bass, L., Clements, P., Kazman, Rick: Software Architecture in Practice, Addison-Wesley Professional, 2ª edicion, 2003.
6. Cantara, M.: Common features of external service providers' SOA frameworks and offerings, Gartner, 2005.
7. Delgado, A., Ruiz, F., García-Rodríguez de Guzmán, I., Piattini, M.: A Model-driven and Service-oriented framework for the business process improvement, Journal of Systems Integration, 2010.
8. Joachim, N., Beimborn, D., Weitzel, T.: Investigating Adoption Determinants of Service-Oriented Architectures (SOA), Proceedings of SIGSVC Workshop, Sprouts: Working Papers on Information Systems, 10(126), 2010.
9. Erl, T.: Service-Oriented Architecture (SOA): Concepts, Technology, and Design, Prentice Hall, 2005.
10. Jones, S.: A Methodology for Service Architectures, Capgemini UK plc, 2005.
11. Arsanjani, A.: Service-Oriented Modelling and Architecture (SOMA), IBM developer-Works, 2004.
12. Huhns, M.N., Singh, M.P.: Service-Oriented Computing: Key Concepts and Principles, IEEE Computer Society, 2005.
13. Zimmermann, O., Gee, K.: Elements of Service-Oriented Analysis and Design, IBM Zurich Research Group, 2004.
14. Zimmermann, O., Schlimm, N., Waller, G., Pestel, M.: Analysis and Design Techniques for Service-Oriented Development and Integration, IBM Deutschland, 2004
15. Caicedo, S.M., Bustos, L.S., Rojas Diaz, J.: Process Integration Making Use of Service Oriented Architecture, Scientia et Technica Año XIV, No 40, 2008.
16. Erradi, A.: SOAF: An architectural framework for service definition and realization, in Proceedings of the IEEE International Conference on Services Computing, pp 151-158, Chicago, USA, September 2006.
17. Erl, T.: Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services, Prentice Hall, 2004.
18. Erl, T.: SOA: Principles of Service Design, Prentice Hall, 2007.
19. Gartner: Magic Quadrant for Enterprise Content Management, 2010.
20. López, G; Echeverría, A.; Fierro, P. (PhD.); Jeder, I. Una propuesta de modelos de ciclo de vida (mcvs) para la integración de los procesos de negocio utilizando service oriented architecture (SOA), Laboratorio de Informática de Gestión Facultad Ingeniería - Universidad de Buenos Aires, 2007

21. Ramollari, Ervin; Dranidis, Dimitris; J. H. Simons, Anthony, A Survey of Service Oriented Development Methodologies, South East European Research Centre, 2007
22. NEXOF-RA, NEXOF Reference Architecture, http://nexof-ra.eu/?q=node/526
23. Pohl, Klaus, S-Cube - The European Network of Excellence in Software Services and Systems, 2012
24. de Castro, Maria Valeria, SOD-M, Aproximación MDA para el desarrollo orientado a servicios de sistemas de información web: del modelo de negocio al modelo de composición de servicios web, Lenguajes y Sistemas Informáticos, Uni. Rey Juan Carlos, 2007

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): Actas de las "VIII Jornadas de Ciencia e Ingeniería de Servicios (JCIS'2012)", Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

Sesión 3

Procesos de Negocio

Chair: *Dr. Inmaculada Medina*

Sesión 3: Procesos de Negocio
Chair: Dr. Inmaculada Medina

Laura Sánchez González, Francisco Ruiz and Félix García. *Guías para el Modelado de Procesos de Negocio*.

Andrea Delgado, Barbara Weber, Francisco Ruiz and Ignacio García-Rodríguez de Guzmán. *A proposal on service execution measures for the improvement of business processes realized by services*.

Clara Ayora, Victoria Torres and Vicente Pelechano. *Feature Modeling to deal with Variability in Business Process Perspectives*.

Adela Del Río Ortega, Cristina Cabanillas Macías, Manuel Resinas Arias de Reyna and Antonio Ruiz Cortés. *PPINOT: A Tool for the Definition and Analysis of Process Performance Indicators*.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): Actas de las "VIII Jornadas de Ciencia e Ingeniería de Servicios (JCIS'2012)", Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

Guías para el Modelado de Procesos de Negocio

Laura Sánchez-González, Francisco Ruiz, Félix García

Instituto de Tecnologías y Sistemas de Información, Universidad de Castilla la Mancha, Ciudad Real, España
{laura.sanchez | francisco.ruizg | felix.garcia}@uclm.es

Resumen. En la etapa de diseño de los procesos de negocio se genera un modelo conceptual. Estos modelos son un artefacto muy útil para detectar errores tempranos y ayudar en la toma de decisiones sobre qué cambios deben ser aplicados para mejorar la eficiencia del proceso. El modelado de procesos de negocio en una organización puede involucrar a un número significativo de participantes sin experiencia, lo que puede llevar a producir modelos de escasa calidad y, en consecuencia, llevar a esfuerzos ineficientes durante el desarrollo y ejecución del proceso. En este trabajo se presentan unas guías para ayudar al modelador a garantizar unos niveles de calidad adecuados. Las guías han sido obtenidas aplicando una serie de pasos, basados en experimentación y técnicas de validación estadística.

Palabras clave: procesos de negocio, guías de modelado, mejora de procesos.

1 Introducción

Las organizaciones prestan cada vez más atención a la mejora de sus procesos de negocio, ya que cuanto más eficiente es una organización, más competente será en el mercado [1]. Un proceso de negocio puede ser visto como una entidad compleja que pasa por diversas etapas, que conforman un ciclo de vida completo. La primera de dichas etapas es el Diseño, cuyo principal valor es disponer de modelos explícitos de los procesos. Aunque no se encuentra entre las fases más costosas en esfuerzos, recursos o costes, puede tener un alto impacto en los beneficios y eficiencia durante la implementación de los procesos [2]. Además, las mejoras incorporadas en los modelos de proceso evitan la propagación de los errores o deficiencias a etapas posteriores, en las cuales la solución suele ser más difícil y costosa [3]. Por estas razones, los modelos de procesos de negocio deben ser diseñados con niveles adecuados de calidad. Sin embargo, al tratarse de una actividad del ámbito de modelado del negocio, es frecuente, al contrario que en diseño de software, que participen actores con poca experiencia en técnicas de modelado, por ejemplo, analistas organizacionales u otros „*stakeholders*" de los procesos. Esta situación supone un riesgo alto de obtener modelos con niveles de calidad no adecuados. Una buena opción es aplicar una serie de guías, consejos o buenas prácticas de modelado. En la bibliografía existen algunas

adfa, p. 1, 2011.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 91-104, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

propuestas en este sentido, pero suelen ser demasiado abstractas o genéricas, perdiendo efectividad práctica, o carecen de fundamentación empírica.

Un paso necesario para mejorar la calidad de cualquier artefacto es la valoración de la misma mediante un adecuado esfuerzo de medición. Así será posible conocer si un modelo satisface un atributo de calidad específico. Existen medidas enfocadas a la calidad estructural de los modelos de procesos de negocio [4], cuya importancia estriba en la validación empírica de su conexión con los atributos de calidad [5], por ejemplo, la entendibilidad y la modificabilidad. La entendibilidad se define como los esfuerzos que los sujetos hacen para reconocer los conceptos lógicos y su aplicabilidad [6], mientras que la modificabilidad se define como el grado con el cual el modelo puede ser efectiva y eficientemente modificado sin introducir defectos o degradar la calidad existente [6]. En procesos de negocio estos atributos de calidad son importantes porque los modelos están continuamente evolucionando para adaptarse a las necesidades cambiantes de las organizaciones.

La medición puede ofrecer información sobre la calidad de los modelos pero, para servir a los modeladores para guiar en la toma de decisiones durante el diseño, es necesario que se exprese en términos comparativos, usando indicadores, respecto de ciertos valores límite o umbrales. Sólo entonces se puede decidir si un modelo es o no adecuado respecto de ciertas propiedades cualitativas. Determinar valores umbral no es una tarea sencilla y prueba de ello es que la gran mayoría de las propuestas sobre medición, de cualquier tipo de artefactos software, no llegan a conseguirlo. Para el caso que nos ocupa, los modelos de procesos de negocio, en trabajos previos hemos obtenido umbrales usando curvas ROC [7], y hemos estudiado medidas (base, derivadas e indicadores), criterios de decisión y valores umbrales para evaluar la entendibilidad y modificabilidad [8].

El marco conceptual subyacente está basado en la „*Software Measurement Ontology*" [9]. Así, para tomar decisiones de diseño no basta con disponer de medidas, base o derivadas, y de sus valores (ej: número de actividades = 25), sino que es necesario definir indicadores basados en dichas medidas para tener valores cualitativos (ej: nivel de complejidad=alto). Los indicadores vienen expresados como una serie de criterios de decisión que establecen el valor cualitativo que corresponde a cada rango de valores de una cierta medida (ej: si número de actividades oscila entre 20 y 40 => nivel de complejidad=alto). Es dentro de los criterios de decisión donde juegan el papel clave los valores umbrales (20 y 40 en el ejemplo anterior). Para completar la cadena que lleva a la toma de decisiones de diseño, es necesario un último eslabón: asociar a los valores cualitativos de cada indicador las guías de diseño que permiten corregir el modelo mejorando el valor del indicador (ej: si nivel de complejidad es alto o muy alto, entonces agrupa actividades en forma subprocesos). Dichas guías de diseño se pueden capturar consultando a expertos (buenas prácticas, patrones), pero sólo mediante experimentación se pueden validar y, sobre todo, se puede establecer su relación pragmática con ciertos valores de ciertos indicadores.

En este trabajo presentamos el método que hemos ideado y las guías e indicadores asociados que hemos obtenido para modelos expresados con el lenguaje BPMN. Para ello, en la sección 2 se describen algunos trabajos relacionados con la medición y con las guías de modelado para modelos de procesos. En la sección 3 se explican los pa-

sos que hemos establecido para definir la guías de modelado, mientras que en la siguiente sección se presenta su aplicación a guías para mejorar la entendibilidad y modificabilidad. Por último, en la sección 5 presentamos las conclusiones y el trabajo futuro de esta investigación.

2 Antecedentes

En esta sección, se describen algunos de los antecedentes necesarios para comprender el trabajo. En primer lugar, se hace un breve resumen de la medición para modelos de procesos de negocio. Después, se comentan algunas de las guías para modelado que se han publicado anteriormente por diversos autores.

2.1 Medición de modelos de procesos de negocio

Gran cantidad de estudios sobre la medición de modelos de procesos de negocio han sido publicados hasta la fecha. En [4], los autores recogieron la mayor parte de las medidas para procesos de negocio en general, y concluyeron que muchas de ellas estaban centradas en los modelos conceptuales de procesos de negocio. De esta manera, se pudo concluir que existe un creciente interés en la medición de los modelos conceptuales. Además, muchas de estas medidas han sido inspiradas por trabajos previos en medidas para el software, como por ejemplo líneas de código, complejidad ciclomática, etc.

La mayoría de las medidas definidas sobre modelos de procesos de negocio están enfocadas en la estructura (flujo de trabajo) de los mismos. Por ejemplo, en [10] los autores proponen contabilizar los elementos más importantes en un modelo, como el número de nodos de decisión, de tareas, etc. para dar una opinión de cómo de compleja es la estructura del modelo de proceso. Estas medidas fueron validadas empíricamente a través de experimentos para comprobar su relación con la entendibilidad y la modificabilidad.

Los experimentos y la validación empírica de medidas han sido el foco de numerosos trabajos. Por ejemplo, Cardoso comprobó la correlación existente de algunas medidas y su capacidad para percibir la complejidad [11]. Otras medidas se definieron basándose en consideraciones cognitivas [12] y en conceptos de modularidad [13]. Otro conjunto de medidas estructurales fueron validadas como predictores de la probabilidad de encontrar errores en los modelos [14]. Otros trabajos demostraron que el tamaño es un factor importante en el modelado a través de medidas tradicionales como el nivel de estructuración del modelo [15]. En conclusión, estas propuestas se dedican a encontrar relaciones de correlación entre las medidas estructurales y aspectos de calidad externa, lo cual permitiría predecir, a partir de la estructura del modelo, cómo de bueno sería el modelo desde el punto de vista de la calidad externa. Estas relaciones están resumidas en la Fig. 1.

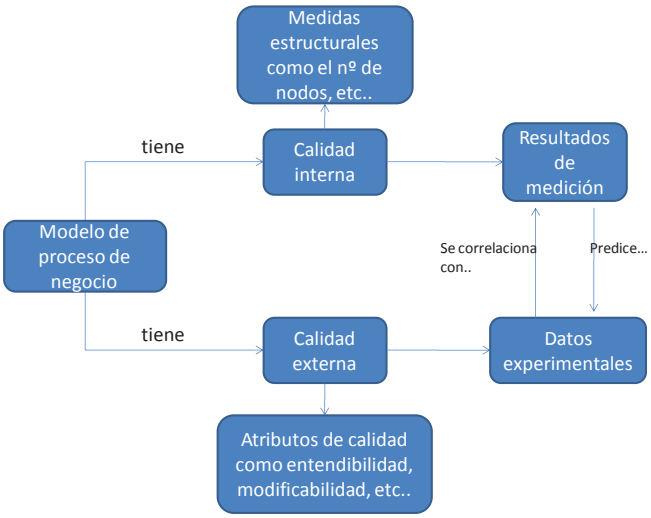


Fig. 1. Relación entre medición y atributos de calidad

2.2 Otras guías para el modelado de procesos

Varios autores han publicado trabajos previos relacionados con guías para el modelado de procesos de negocio. La calidad de los modelos conceptuales ha sido investigada en diferentes marcos como SEQUAL [16]. SEQUAL es un entorno semiótico para la evaluación de modelos conceptuales originalmente propuesto por Lindland et al. [17]. Este marco describe la calidad desde tres perspectivas diferentes: sintaxis, semántica y pragmática, por tanto, los modeladores deben tener en cuenta estas perspectivas para obtener un modelo con una calidad aceptable. Se cree que esta propuesta agrupa las categorías relevantes de calidad, pero resulta muy abstracta y no es directamente aplicable en entornos reales, mucho menos por inexpertos en modelado.

Otro ejemplo de guías de modelado se encuentra en [18], con la propuesta llamada "*the Guidelines of Modeling*". Este marco está enfocado a diferentes usuarios finales, con diferentes objetivos y la posibilidad de diferentes técnicas de modelado y herramientas para describir los modelos. Este marco revela seis técnicas generales para ajustar los modelos a las distintas perspectivas de los diferentes usuarios y objetivos. Estas técnicas están enfocadas a la correctitud, relevancia, economía, eficiencia, claridad, comparabilidad y diseño sistemático. Sin embargo, esta propuesta es también difícil de aplicar por modeladores noveles y no es usada, realmente, en la práctica.

Algunas guías operacionales de modelado de procesos pueden ser encontradas en libros como el de Sharp and McDermott [19]. Sin embargo, la única guía conocida, que defina reglas simples y con fundamentos empíricos es la llamada "Seven Process Modeling Guidelines [20]" y consiste en siete guías de ayuda, como por ejemplo "usar etiquetas lingüísticas *verbo+complemento*" o "modelar de la forma más estructurada posible". El presente artículo pretende extender esta línea de investigación a

través de los valores umbral derivados de técnicas adaptadas a este contexto y medidas estructurales de modelos conceptuales.

3 Pasos para la definición de guías de modelado

En esta sección se describen un conjunto de pasos para definir guías de modelado de procesos de negocio. Los valores umbral, incluidos como parte de la definición de los indicadores, son usados para detectar los elementos de los modelos que deben ser modificados. Algunas veces, estas modificaciones no son triviales y, por eso, las guías pueden resultar muy útiles. Frente a las propuestas comentadas en la sección 2, nuestro valor añadido o particularidad es que las guías o consejos para el modelado están basados y asociados con los valores umbral incluidos en una lista de indicadores. De forma resumida, los pasos para definir las guías son (ver Fig. 2): i) seleccionar un conjunto de medidas base con valores límite asociados, ii) crear una ecuación en la cual la incógnita es la medida (el contador de elementos del modelo) y el resultado el valor límite o umbral, iii) definir una guía de modelado de acuerdo a ese resultado.

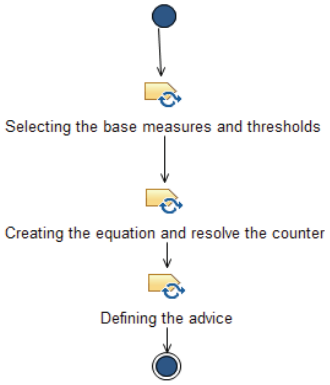


Fig. 2. Pasos para la definición de guías de modelado

4 Guías para mejorar la entendibilidad y modificabilidad

Como ejemplo del método explicado en el apartado anterior, a continuación se presenta su aplicación para mejorar la entendibilidad y modificabilidad.

4.1 Guías para la Entendibilidad

Una de las características de calidad más importantes para los modelos conceptuales es la entendibilidad. Por esta razón, es interesante mantener unos niveles de entendibilidad aceptables en los modelos conceptuales. En el caso de procesos de negocio esto es especialmente importante dado que dichos modelos deben poder ser entendidos por „*stakeholders'* con perfiles muy diferentes.

En trabajos previos de nuestro grupo [21], se realizaron varios experimentos para comprobar la correlación existente entre la entendibilidad y ciertas medidas estructurales de modelos de procesos de negocio representados con BPMN [22]. De estos experimentos se obtuvieron resultados del indicador "eficiencia de entendibilidad". Trabajos posteriores ampliaron el conjunto de medidas estudiadas [8, 23]. Para sacar conclusiones comunes a los diversos experimentos se empleó la técnica de meta-análisis [24], ideada para integrar de forma estructurada y sistemática la información obtenida en diferentes estudios. Los resultados finales sobre correlación de medidas estructurales y entendibilidad se publicó en [25] y se resumen en la Tabla 1 (parte 1 y 2).

Tabla 1. Medidas para entendibilidad y sus valores umbral (parte 1)

Medidas	Muy difícil de entender	Difícil de entender	Moderadamente entendible	Fácil de entender	Muy fácil de entender
Nºnodos	(∞,81.1]	(81.1,58.1]	(58.1,43.7]	(43.7,29.4]	(29.4,6.5]
Diametro	(∞,23.4]	(23.4,16.5]	(16.5,12.2]	(12.2,7.92]	(7.92,1.03]
Densidad	(0,0.06]	(0.06,0.20]	(0.20,0.41]	(0.41, ∞)	-
AGD: nivel medio de nodos de decision	(∞,5.70]	(5.70,3.98]	(3.98,2.90]	(2.90,1.82]	(1.82,0.10]
MGD: nivel máx. de nodos de decisión	(∞ ,8.39]	(8.39,5.3]	(5.3,3.36]	(3.36,1.42]	(1.42,0]
Profundidad	(∞,5.09]	(5.09,3.02]	(3.02,1.72]	(1.72,0.42]	(0.42,0]
GM: Desajuste de los nodos de decisión	(∞,40.9]	(40.9,22.6]	(22.6,11.2]	(11.2,0]	-
GH: heterogeneidad de los nodos de decisión	(∞,1.39]	(1.39,0.71]	(0.71,0.28]	(0.28,0]	-
Secuencialidad	(0,0.25]	(0.25,0.48]	(0.48,0.70]	(0.70,1.07]	(1.07, ∞)
Separabilidad	(0,0.03]	(0.03,0.37]	(0.37,0.71]	(0.71,1.24]	(1.24, ∞)
CNC: coefi. de conectividad	(∞,2.28]	(2.28,1.43]	(1.43,0.90]	(0.90,0.37]	(0.37,0]
TS: token split	(∞,1.36]	(1.36,0.60]	(0.60,0.12]	(0.12,0]	-
CFC: complejidad del flujo de control	(∞,38.2]	(38.2,21.1]	(21.1,10.3]	(10.3,0]	-
NEDDB: nº de join/Split exclusiva basada en datos	(∞,6.02]	(6.02,3.87]	(3.87,2.52]	(2.52,1.17]	(1.17,0]
NEDEB: nº de join/Split exclusiva basada en eventos	(∞,5.76]	(5.76,2.62]	(2.62,0.65]	(0.65,0]	-
NID: nº de join/Split inclusive	(∞,4.63]	(4.63,2.17]	(2.17,0.62]	(0.62,0]	-
NCD: nº de join/Split compleja	(∞,4.56]	(4.56,2.18]	(2.18,0.69]	(0.69,0]	-
NPF: nº de join/Split paralela	(∞,3.36]	(3.36,1.60]	(1.60,0.49]	(0.40,0]	-

Tabla 1. Medidas para entendibilidad y sus valores umbral (parte 2)

Medidas	Muy difícil de entender	Difícil de entender	Moderadamente entendible	Fácil de entender	Muy fácil de entender
NSFG: nº de flujos de secuencia desde nodos de decisión	(∞,42.5]	(42.5,23.2]	(23.2,11.1]	(11.1,0]	-
TNG: nº nodos de decisión	(∞,17.3]	(17.3,9.71]	(9.71,4.89]	(4.89,0.08]	(0.08,0]
NP: nº de participantes	(∞,6.49]	(6.49,4.14]	(4.14,2.66]	(2.66,1.19]	(1.19,0]
PDOPout: proporción de objetos de datos de salida	(∞,1.39]	(1.39,0.79]	(0.79,0.41]	(0.41,0.03]	(0.03,0]
TNE: nº de eventos	(∞,18.2]	(18.2,11.5]	(11.5,7.28]	(7.28,3.04]	(3.04,0]
TNA: nº de actividades	(∞,46.5]	(46.5,31.3]	(31.3,21.8]	(21.8,12.3]	(12.3,0]
TNSF: nº flujos de secuencia	(∞,74.8]	(74.8,50.2]	(50.2,34.8]	(34.8,19.4]	(19.4,0]
CLP: nivel de conectividad entre participantes	(∞,6.32]	(6.32,3.79]	(3.79,2.21]	(2.21,0.62]	(0.62,0]
NDOOut: nº objetos de datos de salida	(∞,19.3]	(19.3,9.60]	(9.60,3.46]	(3.46,0]	-
NDOIn: nº objetos de datos de entrada	(∞, 26.1]	(26.1,12.1]	(12.1,3.38]	(3.38,0]	-
NSFE: nº de flujos de sequencia desde eventos	(∞,16.5]	(16.5,8.74]	(8.74,3.81]	(3.81,0]	-
NMF: nº de mensajes	(∞,22.8]	(22.8,13.2]	(13.2,7.15]	(7.15,1.09]	(1.09,0]

En la literatura se encuentran varias técnicas estadísticas para calcular valores umbral a partir de datos experimentales. En este apartado comentamos uno de los usados en nuestros trabajos, conocido como método Bender [26], ideado inicialmente para el campo de la medicina. Este método permite asociar probabilidades a ciertos resultados de medición del tipo siguiente: si una medida particular *m* obtiene un valor Y ϵ $[Y_1,Y_n]$, entonces existe una probabilidad Z% de considerar el modelo como *muy fácil de entender*. Los resultados de aplicar este método en un conjunto de medidas validadas empíricamente fueron publicados en [8, 27]. Los umbrales asociados a algunas medidas se muestran en la Tabla 1. Además, siguiendo las recomendaciones de [28], se eligieron 5 etiquetas lingüísticas (*muy difícil de entender, difícil de entender, moderadamente entendible, fácil de entender, muy fácil de entender*) para dar una valoración más fácil de entender para el ser humano. A título de ejemplo, supongamos que el número de nodos para un modelo es 70. Entonces, siguiendo la Tabla 1 (primera fila), sabríamos que ese modelo tiene indicios de ser difícil de entender, y por tanto, habría que revisar el número de nodos (el elemento de diseño contado por la medida) del modelo. Los valores umbral constituyen un complemento fundamental para las guías de modelado, ya que pueden ayudar a un modelador a determinar si su modelo

está "bien diseñado" en función de las medidas estructurales. Permiten identificar las circunstancias en que la calidad de un modelo está en peligro. Sin embargo, estos „disparadores" no detallan qué hacer, sólo nos dán el aviso. En el apartado 4.3 se aborda esta cuestión.

4.2 Guías para la Modificabilidad

En esta sección se describen un conjunto de medidas estructurales correlacionadas con la modificabilidad y sus umbrales asociados.

La validación empírica de las medidas estructurales y su relación con la modificabilidad se estudió en trabajos previos de manera similar a cómo se ha comentado con la entendibilidad [21]. Con los experimentos realizados se validaron empíricamente un conjunto de medidas estructurales [8, 23] y se hallaron conclusiones globales mediante meta-análisis [25]. Sin embargo, al contrario que ocurrió con la entendibilidad, el número de medidas capaces de predecir la modificabilidad es más reducido, como se muestra en la Tabla 2.

Tabla 2. Medidas para la modificabilidad y sus valores umbral

Medidas	Muy difícil de modificar	Difícil de modificar	Moderadamente modificable	Fácil de modificar	Muy fácil de modificar
AGD: nivel medio de nodos de decisión	(∞,7.65]	(7.65,4.80]	(4.80,3.02]	(3.02,1.23]	(1.23,0]
MGD: nivel máx. de nodos de decisión	(∞,8.95]	(8.95,5.70]	(5.70,3.66]	(3.66,1.63]	(1.63,0]
GH: heterogeneidad de los nodos de decisión	(∞,1.54]	(1.54,0.81]	(0.81,0.35]	(0.35,0]	-
Separabilidad	(0,0.16]	(0.16,0.42]	(0.42,0.68]	(0.68,1.10]	(1.10, ∞)
NSFG: nº de flujos de secuencia desde nodos de decisión	(∞ ,33.1]	(33.1,18.4]	(18.4,9.26]	(9.26,0.05]	(0.05,0]
CFC: complejidad del flujo de control	(∞,50.6]	(50.6,26.9]	(26.9,12.1]	(12.1,5]	(5,0]
GM: desajuste de los conectores	(∞,42]	(42,24]	(24,12]	(12,1]	-
TNG: nº de nodos de decisión	(∞,15.4]	(15.4,8.56]	(8.56,4.23]	(4.23,0]	-
Profundidad	(∞,5.99]	(5.99,3.26]	(3.26,1.56]	(1.56,0]	-

Los valores umbral para la modificabilidad también se construyeron, en este caso, mediante el método Bender. Los resultados son los mostrados en la Tabla 2. De esta manera, por ejemplo, si la medida GH recibe un valor de 1, el modelo es considerado como *difícil de modificar*. Las guías para solucionar los niveles bajos de calidad se comentan a continuación.

4.3 Guías de Modelado

En esta sección se presentan un grupo de guías que pueden ser usadas para mejorar la entendibilidad y modificabilidad de los modelos. Estas guías permiten resolver los niveles no adecuados de calidad indicados por ciertas medidas (ver tablas 1 y 2) disparadas. Se describen agrupándolos en forma de guías. Estas guías deben ser usadas cuando los resultados de medición son clasificados como *moderadamente entendible/modificable.* Debemos recordar que los valores umbral son diferentes para cada característica de calidad (entendibilidad o modificabilidad). Un resumen de toda la propuesta se muestra en la tabla 3, donde la columna „explicación" resume en forma textual las situaciones no deseables, extraídas de la colección de medidas y valores umbral, y la guía a aplicar en su caso.

G1. Modulariza el modelo a través del uso de subprocesos. Elimina actividades obvias o fusiona actividades con un nivel bajo de granularidad. Recoloca actividades desde el modelo principal a los subprocesos o vice-versa. Las medidas implicadas en esta guía son el número de nodos, TNA, diámetro, y TNSF. TNA está relacionada con el elemento más común en un modelo, la actividad, y está directamente afectada cuando se reduce la medida número de nodos. En esta línea, el diámetro es también afectado, porque la reducción del número de nodos afecta al camino entre el nodo inicial y algún nodo final. Finalmente, si hay un valor bajo de nodos, los flujos de secuencia también se ven reducidos. A número grande de nodos es un problema típico en muchos de los casos. Existen varias soluciones a este problema, como es la modularización (tal y como se describió en [20]). Cuando un modelo tiene un número grande de nodos, es interesante agrupar algunos de ellos y crear un subproceso. Sin embargo, algunos modeladores inexpertos pueden caer en el error de diseñar modelos con un nivel muy bajo de granularidad y poner actividades muy simples u obvias y por tanto, pueden ser eliminadas sin pérdida de información significativa. Finalmente, algunos subprocesos pueden tener varias actividades en común, lo que significa que éstas deben ser recolocadas en un modelo superior (o padre). Estas soluciones pueden ayudar a mejorar los resultados de ciertas medidas estructurales.

G2. Intentar incluir sólo un nodo de inicio y un nodo de fin por participante. Las medidas relacionadas con esta guía son TNE y NSFE. El número de eventos (de comienzo, intermedios o finales) en el modelo directamente afecta a la suma de los flujos que parten o surgen de los eventos. Las soluciones propuestas están basadas en [20].

G3. Eliminar los participantes representados como cajas negras cuando no incluyen información relevante. Las medidas relacionadas con esta guía son NP y CLP, porque ambas están relacionadas con el elemento participante de los modelos. Una solución posible a este problema es eliminar los participantes que están representados como cajas negras en el modelo. La especificación de este tipo de participantes algunas veces implica información redundante. Por ejemplo, cuando una actividad envía un mensaje a un participante representado como caja negra, la información

sobre quién recibe el mensaje puede ser especificada en la propia actividad en vez de en el participante.

G4. Intentar dividir un nodo de decisión con un número alto de flujos de salida en varios nodos de decisión anidados cuando sea posible. Este límite es indicado por las medidas AGD, MGD, y CNC, porque están relacionadas con el número de entradas y salidas a los nodos, principalmente, a los de decisión. Esta solución consiste en separar un nodo de decisión en varios para analizar una pregunta compleja en varios pasos. Esto facilitará el análisis de las tareas, pero puede a su vez incrementar el número de nodos del modelo. El uso de esta guía se restringe, entonces, a las situaciones en las que las medidas relacionadas no se ven perjudicadas en gran medida.

G5. Intentar fusionar varios nodos de decisión cuando las decisiones especificadas en los nodos de decisión están relacionadas. Evitar los nodos OR-split cuando sea posible. Este límite está indicado por las medidas TNG, CFC y GH¨. Todas estas medidas están relacionadas con los nodos de decisión y, por ejemplo, un incremento de la medida TNG puede incrementar la medida CFC y GH. Esta solución está basada en la idea de reducir el número total de nodos de decisión. El problema es que no es posible eliminar un nodo de decisión en un modelo sin que haya pérdida de información, por eso es mejor fusionar algunos de ellos cuando están relacionados, y por tanto, las preguntas simples se unirán en una pregunta más compleja. Es importante evitar como sea posible, el número de nodos OR-split porque incrementa en gran medida el valor de las medidas asociadas.

G6. Usar los patrones de diseño para evitar desajuste en los nodos de decisión. Esta guía enfatiza la importancia de modelar de forma estructurada, especialmente con los nodos de decisión. Patrones relacionados con acompañar a cada *split* un nodo de decisión de tipo *join* y similares fueron publicados en [29]. Esta guía principalmente ayuda a la sincronización de tareas.

5 Conclusiones y trabajo futuro

En este artículo se han presentado unas guías para ayudar al modelado de procesos de negocio. Estas guías han sido construidas a partir de medidas estructurales y sus correspondientes valores umbral. Las medidas fueron previamente validadas empíricamente para comprobar su correlación con la entendibilidad y modificabilidad de los modelos. Los valores umbral se obtuvieron mediante el método Bender, procedente del campo de la medicina y adaptado a esta investigación. De esta manera, cuando un conjunto de medidas superan determinados valores críticos, los modeladores pueden ser avisados para modificar el modelo de acuerdo a unas guías incluidas también en este artículo.

Tabla 3. Guías para el modelado de procesos de negocio

Característica	Medidas	Explicación	Guías
Entendibilidad	Nodos, TNA, Diámetro, TNSF	No usar más de 58 nodos en general, y 31 actividades. El camino más largo entre un nodo de comienzo y uno de fin no debe ser mayor a 16 nodos. No usar más de 50 flujos de secuencia.	**G1:** Modulariza el modelo a través del uso de subprocesos. Elimina actividades obvias o fusiona actividades con un nivel bajo de granularidad. Recoloca actividades desde el modelo principal a los subprocesos o vice-versa.
	TNE, NSFE	No usar más de 11 eventos y no más de 9 flujos de secuencia desde un evento.	**G2:** Intentar incluir sólo un nodo de inicio y un nodo de fin por participante
	NP, CLP	No usar más de 4 participantes y un CLP no debe exceder 3.79	**G3:** Eliminar los participantes representados como cajas negras cuando no incluyen información relevante.
	AGD, MGD, CNC	No usar más de 4 flujos de secuencia de entrada o salida desde un nodo de decisión y 2 por nodo, con un máximo de 5.	**G4:** Intentar dividir un nodo de decisión con un número alto de flujos de salida en varios nodos de decisión anidados cuando sea posible.
Modificabilidad	AGD, MGD	Do not use more than 5 input/output sequence flows from each gateway, with a maximum value of 6.	
Entendibilidad	CFC, TNG, GH	No usar más de 10 nodos de decisión, con una heterogeneidad de no más de 0.71. La medida CFC no debe ser mayor a 21.	**G5:** Intentar fusionar varios nodos de decisión cuando las decisiones especificadas en los nodos de decisión están relacionadas. Evitar los nodos OR-split cuando sea posible.
Modificabilidad	TNG, CFC, GH	No usar más de 9 nodos de decisión, con una heterogeneidad de no más de 0.81. La medida CFC no debe ser mayor a 27	
Entendibilidad	GM	GM no debe ser mayor a 23.	**G6:** Usar los patrones de diseño para evitar desajuste en los nodos de decisión.
Modificabilidad	GM	GM no debe ser mayor a 24.	

Estas guías de modelado constituyen un punto de partida para mejorar el modelado de procesos de negocio en una organización. Las guías han sido diseñadas con fundamentos empíricos, lo que implica una mayor certidumbre sobre su utilidad práctica. Sin embargo, no todos los aspectos a mejorar en un modelo conceptual están relacionados con la estructura, y esto constituye la principal limitación del trabajo. Otros aspectos como las etiquetas asociadas a los elementos en el modelo (por ejemplo, los nombres de las tareas o las cuestiones de los nodos decisión) afectan muy directamen-

te a la entendibilidad y modificabilidad del modelo, y serán estudiados en trabajos futuros. Finalmente, aunque la entendibilidad y modificabilidad están entre las características de calidad más relevantes para los modelos conceptuales, también serán estudiadas otras en trabajos futuros, y así disponer de unas guías de modelado más completas.

Agradecimientos

Este trabajo ha sido parcialmente financiado por los siguientes proyectos: ALTAMIRA (Junta de Comunidades de Castilla La Mancha, Fondo Social Europeo, PII2I09-0106-2463), INGENIOSO (Junta de Comunidades de Castilla La Mancha, PEII11-0025-9533) y PEGASO/MAGO (Ministerio de Ciencia e Innovación y Fondo Europeo de Desarrollo Regional FEDER, TIN2009-13718-C02-01)

Referencias

1. Pfleeger, S.L., *Integrating Process and Measurement. In A. Melton (Ed.), Software Measurement.* International Thomson Computer Press, 1996: p. 53-74.
2. Rosemann, M., *Potential pitfalls of process modeling: part a.* Business process Management Journal, 2006. **12**(2): p. 249-254.
3. Wand, Y. and C. Weber, *Research commentary: Information systems and conceptual modeling–a research agenda.* Info. Sys. Research, 2002. **13**(4): p. 363--376.
4. Sánchez-González, L., F. García, F. Ruiz, and M. Piattini, *Measurement in Business Processes: a Systematic Review.* Business process Management Journal, 2010. **16**(1): p. 114-134.
5. Zelkowitz, M. and D. Wallace, *Experimental models for validating technology.* IEEE Computer, Computing practices, 1998.
6. ISO/IEC, *9126-1, Software engineering - product quality - Part 1: Quality Model.* 2001.
7. Mendling, J., L. Sánchez González, F. García, and M. La Rosa, *Thresholds for Error Probability Measures of Business Process Models.* International Journal of Systems and Software, 2012. **85**(5): p. 1188-1197.
8. Sánchez-González, L., F. García, J. Mendling, and F. Ruiz, *Quality Assessment of Business Process Models Based on Thresholds.* CoopIS 2010 - 18th International conference on Cooperative Information Systems, 2010: p. 78-95.
9. García, F., M. Bertoa, C. Calero, A. Vallecillo, F. Ruiz, M. Piattini, and M. Genero, *Towards a Consistent Terminology for Software Measurement.* Information and Software Technology, 2005. **48**: p. 631-644.
10. Rolón, E., F. García, and F. Ruiz, *Evaluation Measures for Business Process Models.* Simposioum in Applied Computing SAC06, 2006.
11. Cardoso, J., *Process control-flow complexity metric: An empirical validation.* SCC '06: Proceedings of the IEEE International Conference on Services Computing, 2006: p. 167--173.

12. Vanderfeesten, I., H.A. Reijers, J. Mendling, W.M.P. van der Aalst, and J. Cardoso, *On a Quest for Good Process models: the Cross Conectivity Metric*. International Conference on Advanced Information Systems Engineering, 2008.
13. van der Aalst, W.M.P. and K.B. Lassen, *Trasnalting unstructured workflow processes to readable BPEL: theory and implementation.* Information and Software Technology, 2003. **50**(3): p. 131-159.
14. Mendling, J., *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*. 2008: Springer Publishing Company, Incorporated.
15. Laue, R. and J. Mendling, *Structuredness and its Significance for Correctness of Process Models.* Information Systems and E-Business Management, 2009.
16. Krogstie, J., G. Sindre, and H. Jorgensen, *Process models representing knowledge for action: a revised quality framework.* Eur. J. Inf. Syst., 2006. **15**(1): p. 91-102.
17. Lindland, O.I., G. Sindre, and A. Solvberg, *Understanding Quality in Conceptual Modeling.* IEEE Software, 1994. **11**(2): p. 42-49.
18. Becker, J., M. Rosemann, and C. von Uthmann, *Guidelines of Business Process Modeling*, in *Business Process Management*. 2000, Springer Berlin / Heidelberg. p. 241-262.
19. Sharp, A. and P. McDermott, *Workflow Modeling: Tools for Process Improvement and Application Development.* Artech House Publischers, 2001.
20. Mendling, J., H.A. Reijers, and W.M.P. van der Aalst, *Seven Process Modeling Guidelines (7PMG).* Information and Software Technology, 2010. **52**(2): p. 127-136.
21. Rolón, E., F. García, F. Ruiz, M. Piattini, C.A. Visaggio, and G. Canfora, *Evaluation of BPMN Models Quality. A Family of Experiments.* ENASE - International Conference on Evaluation of Novel Approaches to Software Engineering, 2008.
22. OMG. *Business Process Model and Notation (BPMN), Version 2.0*. 2011; Available from: http://www.omg.org/spec/BPMN/2.0/.
23. Rolón, E., J. Cardoso, F. García, F. Ruiz, and M. piattini, *Analysis and Validation of Control-Flow Complexity Measures with BPMN Process Models.* The 10th Workshop on Business Process Modeling, Development, and Support, 2009.
24. Glass, G.V., B. McGaw, and M.L. Smith, *Meta-Analysis in Social Research.* Sage Publications, 1981.
25. Sánchez-González, L., F. García, F. Ruiz, and M. Piattini, *Validación Global de Medidas para Modelos Conceptuales de Procesos de Negocio mediante Meta-Análisis.* Jornadas en Ingeniería del Software y Bases de Datos, 2010: p. 293-298.

26. Bender, R., *Quantitative Risk Assessment in Epidemiological Studies. Investigating Threshold Effects.* Biometrical Journal, 1999. **41**(3): p. 305-319.
27. Sánchez-González, L., F. Ruiz, F. García, and J. Cardoso, *Towards Thresholds of Control Flow Complexity Measures for BPMN Models.* 26th Symposium On Applied Computing SAC 10, 2011: p. 1445-1450.
28. Miller, G.A., *The magical number seven or minus two: some limits on our capacity of processing information.* Psychological Rev, 1956. **63**: p. 81-97.
29. van der Aalst, W.M.P., A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros, *Workflow Patterns.* Distributed and Parallel Databases, 2003. **14**(1): p. 5-51.

A proposal on service execution measures for the improvement of business processes realized by services

Andrea Delgado[1], Barbara Weber[2], Francisco Ruiz[3], Ignacio García-Rodríguez de Guzmán[3]

[1]Computer Science Institute, Faculty of Engineering, University of the Republica, Julio Herrera y Reissig 565, 11300, Montevideo, Uruguay
adelgado@fing.edu.uy
[2]Quality Engineering Research Group, Computer Science Institute, University of Innsbruck, Technikerstraße 21a, 6020, Innsbruck, Austria
barbara.weber@uibk.ac.at
[3]Institute of Information Technologies and Systems, University of Castilla – La Mancha, Camino de Moledores s/n, 13051, Ciudad Real, Spain
{francisco.ruizg, ignacio.grodriguez}@uclm.es

Abstract. The realization of business processes (BPs) by means of services provides the basis for separating their definition from the technologies that implement them. Services can implement an activity, a sub-process or a complete BP, and can be integrated easily into the BP execution without the interoperability problems that had to be solved formerly for systems to achieve integration. A key aspect for the improvement of BPs is to measure their real execution to assess whether they are performing as expected, including the services realizing them. We have defined a BP Execution Measurement Model (BPEMM) in the context of MINERVA framework for the continuous improvement of BPs, which provides execution measures for BPs implemented by services. In this paper we present our vision for the measurement of services execution -for internal and external services- invoked from BPs.

Keywords: business process/services improvement, execution measurement

1 Introduction

"Measurement is the first step that leads to control and eventually to improvement. If you can't measure something, you can't understand it. If you can't understand it, you can't control it. If you can't control it, you can't improve it." [1]. This business principle defines the importance of measurement for improvement. An improvement effort has to support the identification of process deficiencies, for which measures for the execution of business processes (BPs), their activities, performance, resources, cost and results have to be defined, implemented, collected and analyzed on a regular basis. It is not enough to provide measures and the means to analyze them, including tool support, it is also essential to align measures with business strategy and business goals for the entire organization, with the ones that are specific to each BP. This will allow interpreting the information collected from their execution correctly.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 105-110, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

The Business Process Management (BPM) [2] [3] [4] paradigm has been used by the business discipline to define, manage, optimize and improve its BPs for many years now, supporting the BP lifecycle [2]. The realization of BPs by means of services with the Service Oriented Computing (SOC) paradigm [5] provides the basis for separating their definition from the technologies implementing them, supporting an horizontal vision of the organization. Moreover, it helps provide a better response to changes in either the definition and implementation of BPs, with minimum impact on the other by adding an intermediate service layer between them, which allows organizations to introduce changes in a more agilely way [6] [7].

MINERVA [8] framework is focused on the continuous improvement of BPs realized by services with a model-driven approach, based on their execution measurement. To provide support for the latter, we have defined a Business Execution Measurement Model (BPEMM) providing execution measures for BPs and services, and a process to guide the improvement effort, initially defined in [9]. In this paper we present our vision for the measurement of services execution, organized as follows: in section 2 the services execution view of BPEMM is presented, in section 3 the tool support for BPEMM execution measures analysis is shown. In section 4 related work is presented and in section 5 conclusions and future work are discussed.

2 Services execution view of BPEMM

BPEMM has been defined based on the Goal, Question, Metrics (GQM) [10] paradigm, the primary focus being first on understanding BPs business goals (i.e. performance, results, costs, etc.) and then measuring their business results against these goals. Following the GQM, several Goals are defined for the business and BPs with associated questions and execution measures, which are specified using the Software Measurement Ontology (SMO) [11] with base, derived and indicator measures. BPEMM is organized in a tridimensional way: Execution views (Generic BP, Lean BP and Services), "Devil's quadrant" dimensions (time, cost, flexibility and quality) [12] [13] and Granularity levels (activity instances, BP cases, BP). The service view contains measures regarding the execution of services realizing BPs, taking into account the Quality of Services (QoS) requirements for this type of software. Services measures are based on quality attributes such as: performance (i.e. response time: processing time and latency, throughput, capacity), security (i.e. confidentiality, integrity), dependability (i.e. availability, reliability), as defined mainly in [14] [15] [16]. Due to space limitations, in the following we present as an example execution measures for the time and quality dimensions.

2.1 Time dimension

To calculate the measures corresponding to the services execution view in the time dimension we have defined six times of interest for the activities and services execution to be registered in the BP execution. In the first place, in the BP activity we log the defined enabled time (t_1), the start time (t_2) corresponding to when a service is invoked, and the complete time (t_6) corresponding to when the service returns an

answer after processing the request. This is done in the same way as if they were times for manual activities, the only difference being that the resource executing it is a service. In the second place, in the service itself or in the infrastructure executing it, we need to log the times in which it received the invocation (t_3), starts its execution (t_4), and completes its execution (t_5), sending the result to the BP. These defined times are shown in Fig. 1. When we do not have the data on services execution, we can nevertheless use the times registered from the point of view of the BP, to obtain bounds for the response time, which includes communication latency, as well as latency and processing times. In Table 1 services execution measures defined for the time dimension - Response Time are shown.

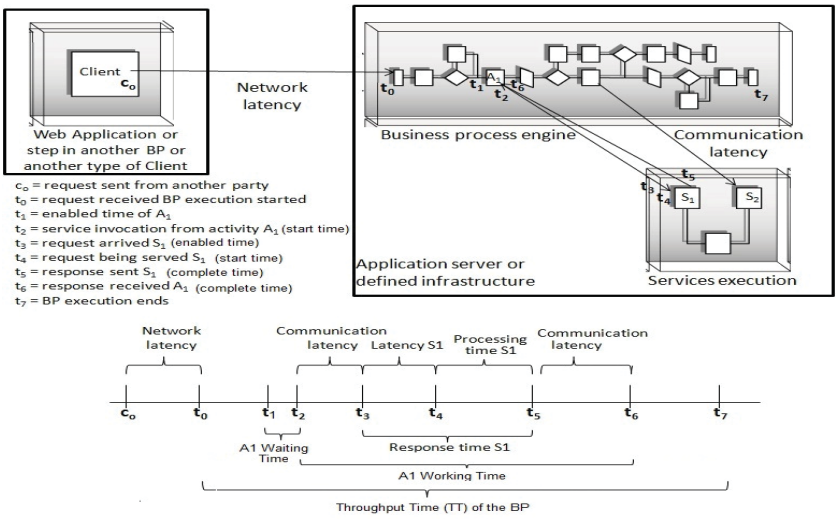


Fig. 1. Defined times for services execution and BP activities

Table 1. Measures for Service execution view & time dimension - Response Time

Goal	G1	**Guarantee (average) service response time to (L1) seconds (L1 label to be changed)**
Questions	Q1	**What is the actual (average) response time of the service**
Measures	M1 (base)	Invoke time of a service from BP (IT=timestamp in activity=ST)
	M2 (base)	Enabled time of a service (ET = timestamp in service)
	M3 (base)	Start time of a service (ST = timestamp in service)
	M4 (base)	Completion time of a service (CT = timestamp in service)
	M5 (base)	Failed time of a service (FT = timestamp in service)
	M6 (base)	Answer time from a service to the BP (AT=timestamp in activity)
	M7 (derived)	Service processing time (SPoT = CT - ST)
	M8 (derived)	Service latency time (SLaT = ST - ET)
	M9 (derived)	Service response time (SRpT = SPoT + SLaT)
	M10 (derived)	Service answer time from the BP (SAnT = AT - IT)
	M11 (indicator)	Service Processing time vs. Service Latency time index (STI = SLaT/SPoT) Decision criteria = Index DC

Goal	G1	**Guarantee (average) service response time to (L1) seconds (L1 label to be changed)**
	M12 (indicator)	Average service response time in all BP cases (ASRpT=∑(SRpT/
	M13 (indicator)	Total service executions all BPs) Decision criteria=Index DC Average service answer time in all BP cases (ASAnT=(SAnT /Total service executions all BPs) Decision criteria=Index DC
Decision criteria	Index DC	R1: 0<=TTI<= L1="LOW"=GREEN; R2: L1<= TTI< L2="MEDIUM"=YELLOW; R3: L2<= TTI="HIGH"=RED

2.2 Quality dimension

Most of the services execution measures data can only be gathered by implementing the log of the base measures in the services and/or the infrastructure, which is true for all quality attributes in this dimension, for example dependability. In Table 2 an example of service execution measures defined for the quality dimension regarding services availability and reliability of the dependability quality attribute.

Table 2. Measures for Service execution view & quality dimension – Dependability

Goal	G1	**Guarantee (A1) availability for the service (A1 label to be changed) - Dependability**
Questions	Q1	**What is the actual availability of the service**
Measures	M1 (derived)	Service downtime (SDT = ET - FT being ET the time when the service is back up, i.e. enabled again after failing)
	M2 (derived)	Total service downtime over the period P1 (TSDT=∑SDT in P1) Service Availability over the period P1(SA=(P1-TSDT)/ P1*100)
	M3 (indicator)	Decision Criteria = Percentage SR DC
Goal	G2	**Guarantee (R1) reliability for the service (R1 label to be changed) – Dependability**
Questions	Q1	**What is the actual reliability of the service**
Measures	M1 (base)	Number of service execution initiated in period P1 = NSEIP
	M2 (indicator)	Service Reliability (SR = NSECP/NSEIP*100) Decision criteria = Percentage SR DC (NSECP = service exec completed in P1)
Decision criteria	Percentage SR DC	R1:0<=TTI<=L1="LOW"= RED; R2: L1<= TTI< L2="MEDIUM"=YELLOW; R3:L2<= TTI="HIGH"=GREEN

3 Tool support for BPEMM execution measures analysis

BPs execution analysis is a key activity if we want to be able to find improvement opportunities for the BPs and the services realizing them, based on the measurement results from their execution. The ProM framework [17] applies process mining techniques on execution event logs to analyze BPs execution, for which we have developed our own prototype plug-in to support the BPEMM execution measures proposal, the ProM BPEMM plug-in. It provides support for the analysis of BPs and services execution, calculating and visualizing the execution measures in BPEMM. We have implemented for now only the time dimension of the BP Generic execution

view, so only the bounds for the service execution (as presented in section 2.1) can be analyzed from the BP activities point of view. This is shown in Fig. 2 for the activity "Receive request appointment" of the BP "Patient Major Ambulatory Surgery (MAS)" from the Hospital General de Ciudad Real (HGCR), which is implemented by a service, showing the waiting, working and total execution time in each BP case.

Fig. 3. ProM BPEMM plug-in time measures for the activity invoking a service

4 Related work

Several approaches exist for measuring services QoS characteristics, where several quality attributes and definitions are put together in [14], and a taxonomy for each one is defined. In [15] these quality attributes are analyzed in the context of a Service Oriented Architecture (SOA) providing specific insight into several characteristics that have to be taken into account when assessing services execution. [16] proposes several measures for QoS for Web Services (WS). Our work integrates existing concepts and execution measures mainly from the proposals above, adapting them for services realizing BPs, to provide a complete view on BPs execution for both manual and automated activities invoking services. Several other works have been analyzed but due to space limitations we have selected the most important for us.

5 Conclusions and future work

The BPEMM execution measurement model provides a set of measures for the execution of BPs realized by services, with a tridimensional organization based on three execution views: Generic BP, Lean and Services; the "Devil's quadrant" dimensions of time, cost, quality and flexibility and a three hierarchy level of

execution (activity instances, BP cases and BP). We have presented here the ones corresponding to the services execution view, providing our vision on the execution of BPs realized by services, which allow us to measure services execution both from the point of view of the BP invoking services, and within services themselves, when they are internal to the organization. This provides a complete view on the execution of BPs and their implementation with services, helping to find improvement opportunities, which are then integrated to generate a new version of the BP following the improvement activities we propose (not presented here). We have developed a prototype for the calculation and visualization of BPEMM execution measures for the ProM framework, the ProM BPEMM plug-in, and we are now extending it to include all the defined measures, to be able to assess the complete proposal in a case study.

Acknowledgments. This work has been partially funded by the Agencia Nacional de Investigación e Innovación (ANII, Uruguay), ALTAMIRA project (Junta Comunidades Castilla-La Mancha, Spain, FSE, PII2I09-0106-2463), PEGASO/MAGO project (Ministerio de Ciencia e Innovación MICINN, Spain, FEDS FEDER, TIN2009-13718-C02-01), INGENIOSO project (Junta Comunidades Castilla-La Mancha, Spain, PEII11-0025-9533) and MOTERO project (Junta Comunidades Castilla-La Mancha, Spain,PEII11-0366-9449).

References

1. Harrington HJ. Business Process Improvement: The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness, McGraw-Hill, (1991)
2. Weske M. BPM Concepts, Languages, Architectures. Springer, (2007)
3. van der Aalst WMP, ter Hofstede A, Kiepuszewski B, Barros A. Workflow Patterns. Distributed and Parallel Databases. Vol. 14, Issue 3:5 – 51, (2003)
4. Smith H, Fingar P. Business Process Management:The third wave. Meghan-Kieffer; 2003.
5. Papazoglou M, Traverso P, Dustdar S, Leymann F. Service-Oriented Computing: State of the Art and Research Challenge. IEEE Computer Society, (2007)
6. Erl T. SOA: Concepts, Technology, and Design. Prentice Hall; 2005.
7. Krafzig D, Banke K, Slama D. Enterprise SOA Best Practices. Prentice Hall, (2005)
8. Delgado A, Ruiz F, García-Rodríguez de Guzmán I, Piattini M. MINERVA: Model drIveN and sErvice oRiented framework for the continuous business process improVement & relAted tools. In: 5th Int. Workshop on Engineering SO Applications (WESOA'09),(ICSOC'09) (2009)
9. Delgado A, Weber B, Ruiz F, García-Rodríguez de Guzmán I. Execution Measurement-driven continuous improvement of business processes implemented by services. In: 6th. Int. Conference on Evaluation of Novel Approaches to Software Engineering (ENASE'11), (2011)
10. Basili VR. Software Modeling and Measurement: The GQM Paradigm. University of Maryland, CS-TR-2956, (1992)
11. García F, Bertoa M, Calero C, Vallecillo A, Ruiz F, Piattini M, et al. Towards a Consistent Terminology for Software Measurement. Information and Software Technology,Vol. 48,(2005)
12. Brand N, van der Kolk H. Workflow Analysis and Design, (1995)
13. Reijers H. Design and Control Workflow Proc.: BPM for Service Industry, Springer; (2003)
14. Barbacci M, Klein M, Longstaff T, Weinstock C. Quality Atributtes. Software Engineering Institute (SEI), CMU/SEI-95-TR-021, (1995)
15. O'Brien L, Bass L, Merson P. Quality Attributes and SOA. Software Engineering Insttitue (SEI), CMU/SEI-20055-TN-014, (2005)
16. Sahai A, Ouyang J, Machiraju V, Wurster K. Specifying and guaranteeing QoS for WS through real time measurement and adaptive control, HP Laborataries, Palo Alto, (2001)
17. ProM framework; 2004-2011. http://www.processmining.org/

Feature Modeling to deal with Variability in Business Process Perspectives*

Clara Ayora, Victoria Torres, and Vicente Pelechano

Centro de Investigación en Métodos de Producción de Software
Universitat Politècnica de València
Camino de Vera s/n, 46022 Valencia, Spain
{cayora,vtorres,pele}@pros.upv.es

Abstract. The construction of Business Process (BP) models entails big challenges, especially when BPs contain many variations. In addition, BPs can be seen from different perspectives, i.e., the behavioral (i.e., control-flow), the organizational (i.e., resources distribution), or the informational (data-flow) perspectives among others. Depending on the context where the BP is taken place, we may found variability in any of these perspectives. Different approaches to model variability in BP perspectives have already been proposed. However, these approaches are highly tight to the modeling language. In addition, they focus mainly on the behavioral perspective. To deal with variability in other BP perspectives in a more flexible manner, this work proposes an approach based on feature models. These models do not only allow enhancing expressiveness regarding BP variability, but also the maintenance and understanding of the resulting process model.

Keywords: Business Process Modeling, Business Process Variability, Feature Models

1 Introduction

It is common to find *Business Processes* (BPs) that are shared by many organizations, e.g., buying and selling, merchandise delivery, quality control, packaging BPs. However, such BPs usually need to be adapted depending on the context where they are applied, e.g., a *merchandise delivery process* strongly depends on the goods being delivered as well as on the country where the process takes place. As a result, we can find many variations of the same BP according to the specific needs of the application context.

Furthermore, BPs can be seen from a number of different perspectives [14]. Among others, we can talk about the behavioral perspective (i.e., control-flow definition), the organizational perspective (i.e., resources that perform the activities), or the informational perspective (i.e., data objects produced or manipulated by the process) [8]. In that sense, variations not only may occur in one

* This work has been developed with the support of MICINN under the project EVERYWARE TIN2010-18011.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 111-124, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

of these perspectives, but in all of them and at the same time. For example, in the merchandise delivery process, depending on the people in charge and on the goods being delivered, the process may be different. Thus, enterprises should properly manage variations in every perspective in order to copy with market conditions in an effective manner [20].

However, BP variability management is not a trivial task, specially regarding BP variability modeling [13]. When the modeling task involves representing variations, the complexity of the modeling process increases, turning models into artifacts that are error-prone and complex to build, manage and understand [6].

Coping with such variability modeling issues constitutes one of the challenges currently faced by the BP community [3]. Good examples of such interest are the approaches developed by *Puhlmann et al.* [17], *Hallerbach et al.* [7], *Rosemann et al.* [18], and *Kumar et al.* [11]. These approaches constitute a solid base to face the modeling challenges that arise when dealing with BP variability. However, they are highly tight to the *Business Process Modeling Language* (BPML) they were developed to, reducing their flexibility to be used with other BPMLs. In addition, most of the approaches only focus on specific perspectives (usually the behavioral one), neglecting the representation of variations in the rest of them.

In this context, we present a modeling approach to capture the variability that may appear in BP perspectives. Concretely, we focus on BP perspectives beyond the behavioral one. For such purpose, we rely on *feature models* from the Software Product Line (SPL). Feature models are expressive models designed to represent variations in software systems. They have been proved to be the best mechanism to promote model maintenance and understanding regarding variability [9]. Concretely, we propose to define the variations that refer to BP perspectives as features in the feature model. An important aspect of our approach is that its flexibility since it is language independent, which allows applying it to any BPML. However, for explanatory purposes we have taken the *Business Process Modeling Notation* (BPMN) since it is the standard language for BP definition.

The remainder of the paper is organized as follows. In Section 2, we use a scenario to illustrate the motivation of our approach. Section 3 presents the different BP perspectives that are commonly found. Section 4 first introduces feature modeling and then explains how this technique is used to deal with BP variability modeling. Section 5 presents the compilation of the related work. Finally, Section 6 concludes the paper and outlines the future work.

2 Case Study

To motivate the challenges being faced in this work, we present the check-in process usually offered by airlines. The process represents the procedure that every passenger has to go through when traveling by plane. Even though this process is similar irrespective of the airport the passenger departs from and the airline he/she is flying with, many variations are possible depending on different factors.

The process starts by identifying the passenger who is going to travel. This activity may start 23 or 3 hours before departure depending on the method used by the passenger to do the check-in (i.e. web application, self-servicing machines or airport counter). In addition, this also determines the responsible for doing this activity. For instance, if the check-in is performed online (i.e., through a web application), the identification is done by the passenger him/herself. Otherwise, it is done by the airline personnel. Afterwards, the assignment of the seat starts. If the checking is done at the counter, the seat assignment is done by the airline personnel. On the contrary, the assignment is done automatically by the web application (i.e., online check-in) or the self-servicing machine. As an extra service, some airlines offer the possibility of changing the seat assignment if the passenger has bought a business ticket and he/she does not agree with the first assignation. Again, this activity depends on the type of check-in. In this case, the passenger him/herself can change his/her seat when doing online or self-servicing check-in, while at the counter the change is performed by the airline personnel. The next activity to carry out regards the addition of special information. For instance, when passengers travel to USA, they have to provide specific contact information about the address where he/she will stay the first night. Once all the required information has been provided, the passenger obtains the boarding card, either in an electronic format (i.e., online check-in) or paper format (i.e., check-in at the counter check-in or the self-servicing machine). Finally, the process ends at the airport with the luggage drop off by the airline personnel. Figure 1 shows a simplified version of this process. It includes several *process variants* (*variants* for short) of this process.

As it is shown, despite it is a simple process there might be dozens to up to hundreds of possibles variants depending on, for instance, the type of check-in (i.e., online, counter or self-servicing machine) or the type of ticket (i.e., business or economy class). Figure 2 shows one variant of this process regarding the online check-in of a business ticket. In this variant, the process is mainly performed by the passenger him/herself.

3 Business Process Perspectives

According to [21], a BP is defined as "a set of activities performed in coordination in an organizational and technical environment. These activities jointly realize a business goal.". Considering this definition, a BP defines what (activities) should be done, how (coordination), and by whom (organizational and technical environment). In this context, *business process models* arise as the main artifacts for representing BPs. However, other elements are also used to build a BP model. The complete set of these elements is gathered in the meta-model presented in Figure 3.

Each one of these elements, when considered separately, represents a different perspectives of a BP [4, 14, 10]. Therefore, depending on the object we focus on a distinction can be made between the functional, behavioral, organizational, informational, temporal, and operational perspective:

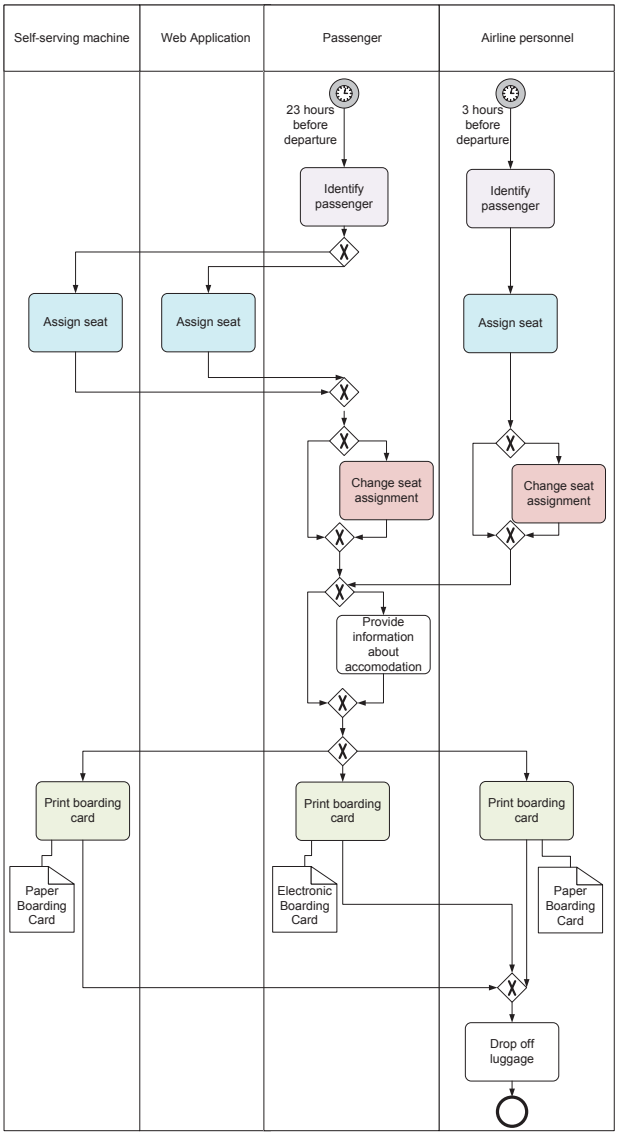


Fig. 1. Check-in process

– **Functional perspective** (specifies the decomposition of processes): It represents the activities to be performed in the context of a BP [4]. This perspective is represented by the elements *Activity*, *Atomic Activity*, and *Complex Activity* from the meta-model (see Figure 3).

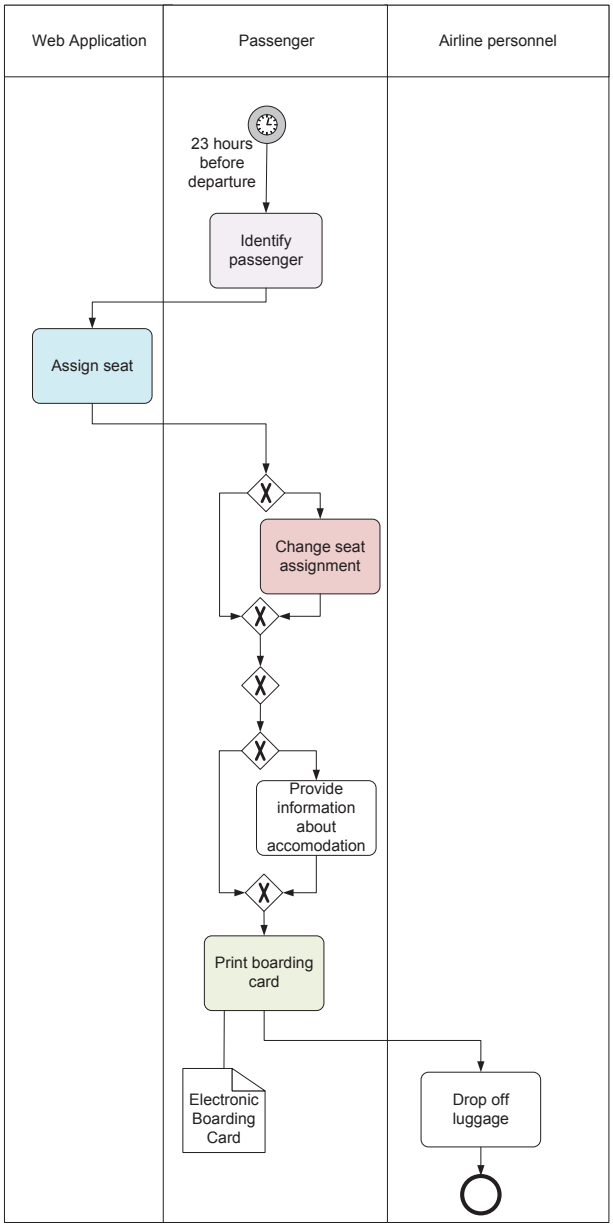


Fig. 2. Variant of the Check-in process regarding the online check-in of a business ticket

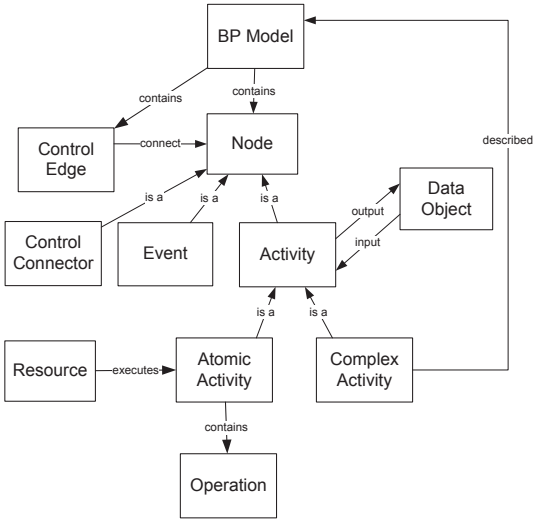


Fig. 3. Business Process Meta-model

- **Behavioral perspective** (defines control-flow): It describes the sequencing of activities and comprises information regarding other constraints for their execution (e.g., sequences, conditional branching, parallel branches, or loop structures). The behavior perspective is represented by the *Control Connector* and *Control Edges* elements from the meta-model.
- **Organizational perspective** (deals with the assignment of resources): It represents by whom in the organization the activities of a process are performed. This perspective is represented by the *Resource* element from the meta-model.
- **Informational perspective** (concerns data and data-flow): It represents the informational entities produced, manipulated or consumed by a process; these entities include data, artifacts, products (i.e., intermediate and end products), and objects; this perspective covers both the structure of informational entities and their relationships. It is represented by the *Data Object* element from the meta-model.
- **Temporal perspective** (deals with time issues and temporal constraints): It represents the occurrence of an event during the course of a process, which affects the scheduling of activities from this process. This perspective is represented by the *Event* element from the meta-model.
- **Operational perspective** (concerns the implementation of process activities): It covers the implementation of the process activities, i.e., the application services to be performed or the forms to be filled in when executing these activities. This perspective is represented by the *Operation* element from the meta-model.

There are still other perspectives associated with a BP model which are not directly related to the BP meta-model elements. These refer to more abstract concepts such as the intended use of the representation [2], or the structural characteristics of BP representations and their capabilities for analysis and optimization [19]. However, these perspectives are out of the scope of this work.

All the perspectives listed above are connected by means of the relationships defined between the elements from the meta-model. In particular, the *Activity* element (that belongs to the functional perspective) articulates the informational, operational, and organizational perspectives. These relationships affect the way variability can be represented regarding these perspectives. For example, an *Activity* that can be performed by more than two different roles (as *Identify passenger*, *Assign seat*, *Change seat*, and *Print boarding card* activities in the check-in process; see Figure 1) has to be represented graphically as many times as different roles can take the responsibility for such *Activity*. Another example is the different types of products (i.e., *Data Objects*) that can be produced by the same *Activity* depending on the role (i.e., *Resource*) taking the responsibility of the activity (e.g., electronic and paper boarding cards produced by the *Print boarding card* activity in the check-in process; see Figure 1). In these two examples, variability occurs in different perspectives (organizational and informational) but related always to an *Activity* element (*Identify passenger*, *Assign seat*, *Change seat assignment*, and *Print boarding card*). This relationship leads to the construction of diagrams as the one presented in Figure 1, where different elements need to be duplicated in order to represent all different variations (see colored activities). However, this solution cannot be applied in practice since it leads to model redundancy, which hinders process model understanding, evolution, and reusability [6]. To solve this situation, in next section we present feature modeling as a mechanism to properly represent variability in all these connected perspectives.

4 Representing Variability in Business Process Perspectives

As it is explained in Section 3, three elements are the cornerstone of BP perspectives: *Control Connector* (for the behavioral perspective), *Event* (for the temporal perspective), and *Activity* (for the functional, organizational, informational, and operational perspectives). Traditionally, variability in the functional and behavioral perspective is easily represented using the primitives provided by traditional BPMLs[1] (e.g., regarding the check-in process, the possible performance of the activity *Change seat assignment* can be simply represented with an XOR gateway; see Figure 1). However, handling variability in the remaining BP perspectives requires a more powerful and flexible technique [12]. This is due to dependency relationships between elements from these perspectives. For example, the check-in process can only be started 23 hours before departure when

[1] It results more intuitive for modelers since only variations in control-flow distribution are involved [18]

the role initiating this process is the passenger him/herself. However, traditional BPMLs do not allow representing such dependency relationships declaratively but implicitly in the model, which brings understanding, evolution, and reusability problems [6].

In this context, feature modeling appears as a good solution to address these problems. The following subsections present first some preliminaries about feature modeling and second our approach based on this technique to model variability in BP perspectives. Concretely, it regards the organizational, informational, temporal, and operational perspectives.

4.1 Feature Modeling: Preliminaries

Feature modeling is a technique developed in the context of the SPL for capturing and describing variability in software systems [9]. Features are variations that are used to differentiate the characteristics of each system. A feature model is represented as a tree-like set of features that are hierarchically arranged with a composed-by relationship between a parent feature and its child features. Figure 4 shows a generic example of a feature model.

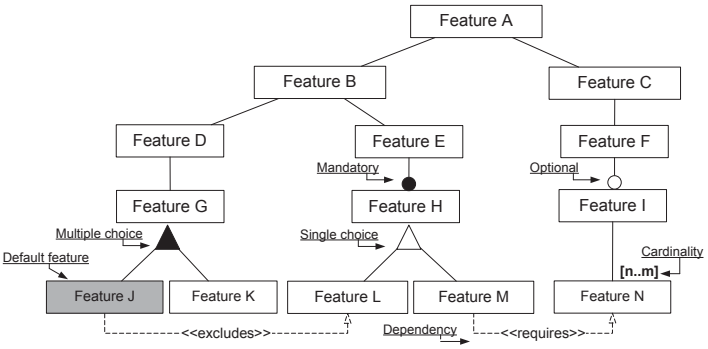


Fig. 4. Generic example of a feature model

In addition there are cross-tree constrains that describe the most typical relationships between features [5]:

1. **Mandatory and Optional**. A mandatory feature must be always included in a system meanwhile an optional feature may or may not be included (see Figure 4).
2. **Multiple choice groups**. It defines a non-empty subset of features that can be included in a system (i.e., OR decisions;).
3. **Single choice**. It indicates from a set of alternative features exactly one feature must be included (i.e., XOR decisions).

4. **Cardinality**. It defines the lower and upper bound values of features that can be included in a system.
5. **Require and Exclude**. A require relationship forces the selection of a feature when another feature has been selected. On the contrary, an exclude relationship prevents the selection of a feature when another feature has been selected.
6. **Default feature**. It represents the feature that is selected in a system by default.

An interesting characteristic of these models is that no structural restrictions among the hierarchical structure are defined. Therefore, the structure of a feature model can be defined as needed depending on the domain being modeled. For example, branches of features can be enlarged or reduced to properly fit system specifications. Thus, a flexible representation of features (i.e., variations) is provided. For these reasons, we propose to exploit this technique to represent variations in BP perspectives.

4.2 Applying Feature Modeling for Business Process Perspectives

Given the flexibility provided by feature models, we propose to organize its tree-like structure in five different levels associated with specific semantics. Thus, a better understanding and management of variability in the different perspectives is provided.

Concretely, the tree-like structure is hierarchically organized based on model elements presenting variability. Then, this variability is also organized according to the perspectives it belongs to. Figure 5 depicts graphically how the feature model has been designed to represent variability in the different perspectives of the check-in example. The different levels and their semantics are explained in the following by using this figure.

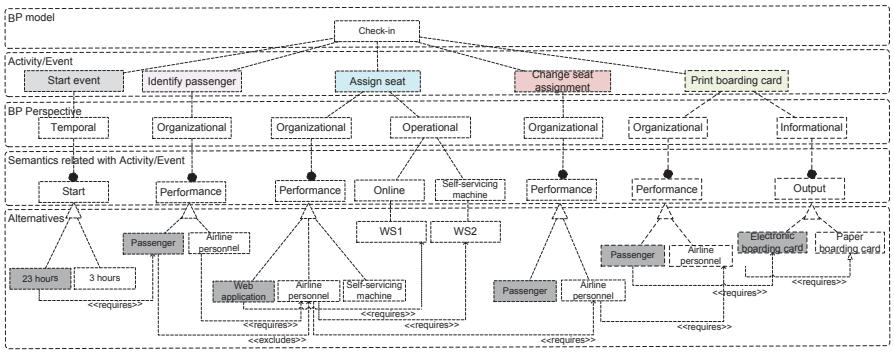


Fig. 5. Feature model associated to the check-in process

Tree-like Structure Organization

The *first level* defines the root feature that is used to derive all variations related to *Activities* and *Events*[2]. As it is shown in Figure 5, the first level includes one feature (i.e., *Check-in*) from which all the tree-like structure is going to derive.

The *second level* defines one feature for each *Activity* (either *Atomic* or *Complex*) or *Event* element from the BP model that presents variability. As it is shown in Figure 5, the second level includes five features that correspond either to an *Activity* (i.e., *Identify passenger, Assign seat, Change seat assignment,* and *Print boarding card*) or an *Event* (i.e., *Start event*) from the check-in process.

The *third level* is used to categorize in perspectives the variability defined by each *Activity* or *Event* defined in the previous level. It includes features for all BP perspectives that present variability in the current process. These features are grouped in the way they can be articulated from the upper level feature. As it is shown in Figure 5, the third level defines for example *Temporal* perspective associated to the *Start event* feature defined in level two. Categorizing features in perspectives facilitates the identification and management of all features.

The *fourth level* defines the semantics of the associated feature defined in the next level. This semantics allows a better understanding of the rationale behind each feature. As it is shown in Figure 5, two different ways to implement the *Assign seat* activity are considered, one by means of an *Web Application* and another one by means of a *Self-servicing machine*.

The *fifth level* defines all the different alternatives for the categories defined in the previous level. These alternatives can be selected (i.e., configured) to derive a process variant. In addition, at this level different relationship between features of the same level can be defined. For example, the 23 hours feature related to the *Start event* defines a relationship with the *Passenger* feature defined for the *Identify passenger* activity. This relationship is establishing a constraint of use between these two features, indicating that the selection of the former enforces the selection of the latter.

Tree Configuration

The complete feature model defined in Figure 5 represents the alternatives of the possible variants that can be derived from the check-in process. In turn, a concrete variant (e.g., variant of the online check-in presented in Figure 2) may be declaratively specified by selecting the desired features according to variant requirements (e.g. check-in online starts 23 hours before departure). Such decisions must respect the cross-tree constraints imposed by the feature model (e.g., the choice of exactly one feature from a group of single choice features). The result of this selection is a *configuration of the feature model*, which includes

[2] Notice that with these two elements we are covering the variability that can occur in all the BP perspectives, e.g., variability regarding the organizational, informational, temporal, and operational perspectives.

the alternatives conforming the concrete variant to be derived. Figure 6 presents the configuration of the feature model associated to the online check-in. By this configuration, variant depicted in Figure 2 is easily derived.

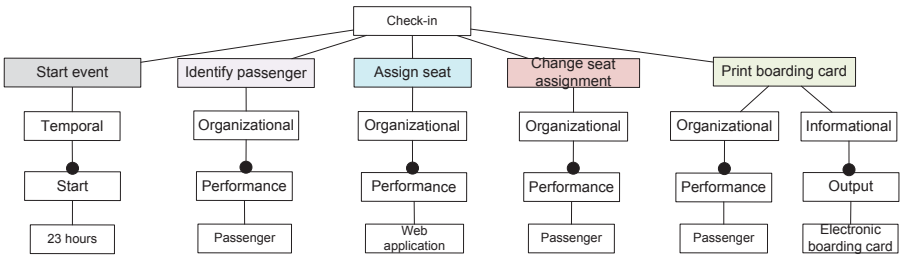


Fig. 6. Feature model configuration associated to the online check-in

5 Related Work

This section describes a short review of approaches from the BP management field dealing with BP variability modeling. These approaches are PESOA [17], Provop [7], C-EPC [18], Rules representation [11], Collaborative activities [16], and Feature models to business processes (FM2BP). They were developed due to the limitation of the original BPML to properly deal with variability in BP modeling. Table 1 summarizes the support each approach provides for the different BP perspectives.

	PESOA	Provop	C-EPC	Rule	Collaborative act.	FM2BP
Functional	+	+	+	+	-	+
Behavioral	+	+	+	+	-	+
Organizational	-	-	+	+	+	-
Informational	-	-	+	+	-	-
Temporal	-	-	-	-	-	-
Operational	-	-	-	-	-	-

Table 1. Summary of the approaches support for the BP perspectives

PESOA [17] provides an approach for the development and customization of families of process-oriented software. PESOA allows describing variation points by attaching annotations to those places (i.e., activities) where variability may occur. For this, strategies like encapsulation, inheritance, design patterns, and

extension points are used. These strategies only refer to the behavioral and functional perspectives. However, alternatives regarding other modeling elements, such as resources or data, cannot be explicitly represented.

Provop [7] describes an operational approach to support BP variability modeling through a set of high-level change operations. Thus, process variants are obtained by applying different change operations to a common base model. However, similarly to PESOA, alternatives can only be defined when talking about control flow options (i.e., behavioral perspective), neglecting variations in other perspectives.

Configurable EPC (C-EPC) [18] is an extension of EPC (Event-driven Process Chain) that includes additional constructs to model variability in reference process models. The *configurable functions* and *connectors* allow identifying which model parts are subject to variations. The approach also support variations in other perspectives such as the organizational and informational perspectives [12]. However, C-EPC is highly tight to EPC which reduces its flexibility to be applied with other BPMLs.

Rule representation and processing [11] is an approach for process configuration based on the idea of configuring variants through a set of business rules. These rules are applied to a generic template, which contains a very basic general process schema. This template constitutes a default configuration that is modified according to context changes. Alternatives are described through a set of business rules, whose application leads to the execution of a set of change operations (e.g., insert or delete activity). These change operations, when applied to the generic process template, lead to a specific process variant. In addition to control-flow, alternatives may also refer to the organizational and informational perspectives. However, alternatives are not explicitly defined for any of those perspectives since the information is scattered among the different rules. In addition, the definition of the alternatives implies to learn specific semantic knowledge.

Collaborative activities [16] is an approach that proposes representing BP collaboration between multiple resources by using colors for each resource instead of the traditional BPMN lanes. The idea is associating roles with colors and using these colors in activities to represent the assignment to a specific role. As a result, when an activity is performed by just one role, the activity will be colored with the corresponding color. On the contrary when the activity can be performed by more than one role, the activity is represented either as many times as roles are involved in the activity (each one with the corresponding color) or as an activity with vertical stripes including the colors that represent the involved roles. Despite this approach was not developed for representing variability in BP perspectives, it allows modeling different alternatives for the organizational perspective. However, the approach neglects variations in other perspectives such as the informational or the temporal perspectives.

Finally, feature models have already been used when dealing with BP variability. For instance, in [15] proposes the use of a feature model (containing the variability) for automatically building a BP. For such construction, a mapping

relationship between the feature model and the control-flow of the BP is defined. Thus, variability specification included in the feature model is transferred to the BP structure (i.e., functional and behavioral perspectives). Apparently, a combination between this approach and our approach allow modeling BP variability in all BP perspective.

6 Conclusions and Further Work

In this paper, we address the lack of support for modeling variability in BP perspectives beyond the behavioral perspective. For such purpose, we have proposed a *feature-oriented* approach to modeling and managing variations in business processes perspectives beyond the behavioral perspective. These models allows defining variations in a more expressive and flexible way. Feature models allow defining not only the variations in the perspectives, but also relationships between them. In addition, the proposed feature model is hierarchically structured, which allows a semantical organization of the features. Consequently, our approach is able to represent alternatives within BP perspectives gaining in model understanding and maintenance. Moreover, feature models are language independent so that they can be applicable to any concrete BPML.

While the approach has been exemplified by the conduction of a case study, further validation is required. This includes applying the approach in other domains, but also designing tool support for configuration and conducting usability testing. For such reason, we are currently developing a prototype tool to assist in the definitions of the feature model. This prototype is been developed in the context of *Eclipse Framework* [22], specifically, it is based on the MOSKitt tool [23]. In particular, we are implementing a graphical editor as an Eclipse plugin in order to integrate it with existing BPMN modeling editors [24]. Thus, scalability and complexity analysis can be rigorously conducted. Specifically, we want to quantify the overhead produced by modeling large feature models. Typically, the manageable size (i.e., number of features) of feature models scales up to 100 features [1]. However, modeling certain features in separate diagrams and/or at different levels of abstractions allow dealing with models over 300 features [1]. Thus, a set of standard benchmarks would be desirable to show how these scalability-issues are managed in practice with our prototype. In addition, we will investigate techniques for preventing inconsistencies and assuring model correctness. Furthermore, this work has been developed as part of a bigger project aimed at supporting variability during the entire BP lifecycle. Thus, we are now defining mechanisms to deal with variability at each phase of the BP lifecycle (i.e, analysis-design, configuration, enactment, diagnosis, and evaluation).

References

1. Acher, M., Collet, P. Lahire, P, France, R.: Comparing Approaches to Implement Feature Model Composition. Europ. Conf. on Model Driven Architecture - Foundations and Applications, 3–19 (2010).

2. Aguilar-Savén, R.S.: Business process modelling: Review and framework. Int. Journal of Production Economics 90(2), 129–149 (2004).
3. Cardoso, J., Mendling, J., Neumann, G., Reijers, H.A.: A discourse on complexity of process models. BPM Workshops, 117–128 (2006).
4. Curtis, B., Kellner, M., Over, J.: Process modeling. Communication of the ACM 35(9), 75–90 (1992).
5. Czarnecki, K., Helsen, S., Eisenecker, U.W.: Formalizing cardinality-based feature models and their specialization. Soft. Proc.: Improvement and Practice, 7–29 (2005).
6. Hallerbach, A., Bauer, T., Reichert, M.: Configuration and management of process variants. International Handbook on Business Process Management, Springer Publisher, 237–255 (2010).
7. Hallerbach, A., Bauer, T., Reichert, M.: Capturing variability in business process models: the Provop approach. Soft. Proc.: Impro. and Prac. 22(6–7), 519–546 (2010).
8. Jablonski, S., Bussler, C.: Workflow Management: Modeling Concepts, Architecture and Implementation. International Thomson Computer Press (1996).
9. Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., Peterson, A.S.: Feature-oriented domain analysis (FODA) feasibility study. TR, Carniege-Mellon Univ.(1990).
10. Korherr, B.: Business Process Modelling: Languages, Goals, and Variabilities. PhD thesis. Institut für Softwaretechnik und Interaktive Systeme, Vienna, Austria (2008).
11. Kumar, A., Wen, Y.: Design and management of flexible process variants using templates and rules. Int. Journal Computers in Industry 63(2), 112–130 (2012).
12. La Rosa, M., Dumas, M., Hofstede, A., Mendling, J., Gottschalk, F.: Beyond control flow: extending business process configuration to roles and objects. Int. Conference on Conceptual Modeling, LNCS vol. 5231, 199–215 (2008).
13. La Rosa, M.: Managing variability in process-aware information systems. PhD thesis. Queensland University of Technology, Brisbane, Australia (2009).
14. Melão, N., Pidd, M.: A conceptual framework for understanding business processes and business process modelling. Information Systems Journal 10(2), 105–130 (2000).
15. Montero, I., Peña, J., Ruiz-Cortés, A.: From Feature Models to Business Processes. IEEE Int. Conference on Services Computing, 605–608 (2008).
16. Müller, R., Rogge-Solti, A.: BPMN for Healthcare Processes. Workshop on Services and their Composition (ZEUS 2011), (2011).
17. Puhlmann, F., Schnieders, A., Weiland, J., Weske, M.: Variability mechanisms for process models. Technical report, BMBF-Project (2006).
18. Rosemann, M., van der Aalst, W.M.P.: A configurable reference modeling language. Inf. Syst. 32(1), 1–23 (2007).
19. Vergidis, K., Tiwari, A., Majeed, B.: Business process analysis and optimization: beyond reengineering. IEEE Transactions on Systems, Man, and cybernetics, 38(1), 69–82 (2008).
20. Weber, B., Sadiq, S.W., Reichert, M.: Beyond rigidity - dynamic process lifecycle support. Computer Science - R&D 23(2), 45–65 (2009).
21. Weske, M.: Business process management: concepts, languages, architectures. Springer-Verlag Berlin Heidelberg Publisher (2007).
22. `http://www.eclipse.org` Accessed: May 2012.
23. `http://www.moskitt.org/` Accessed: May 2012.
24. `http://www.eclipse.org/projects/project.php?id=soa.bpmnmodeler` Accessed: May 2012.

PPINOT: A Tool for the Definition and Analysis of Process Performance Indicators*

Adela del-Río-Ortega, Cristina Cabanillas, Manuel Resinas, and Antonio Ruiz-Cortés

Universidad de Sevilla, Spain
{adeladelrio,cristinacabanillas,resinas,aruiz}@us.es

Abstract. A key aspect in any process-oriented organisation is the evaluation of process performance for the achievement of its strategic and operational goals. Process Performance Indicators (PPIs) are a key asset to carry out this evaluation, and, therefore, having an appropriate definition of these PPIs is crucial. After a careful review of the literature related and a study of the current picture in different real organisations, we conclude that there not exists any proposal that allows to define PPIs in a way that is unambiguous and highly expressive, understandable by technical and non-technical users and traceable with the business process (BP). Furthermore, it is also increasingly important to provide these PPI definitions with support to automated analysis allowing to extract implicit information from them and their relationships with the BP. In this work we present PPINOT, a tool that allows the graphical definition of PPIs together with their corresponding business processes, and their subsequent automated analysis.

1 Introduction

It is increasingly important to evaluate the performance of business processes (BPs), since it helps organisation to define and measure progress towards their goals. Performance requirements on BPs can be specified by means of Process Performance Indicators (PPIs).

According to Franceschini *et al.* [1] and based on the conclusions drawn in our previous works [2, 3], four critical elements for indicators can be identified: (1) their *definition*, that should be unambiguous and complete; (2) *understandability*, PPIs should be understood and accepted by process managers and employees; (3) *traceability with the BP*, enabling to maintain coherence between both assets, BP models and PPIs; and (4) *possibility to be automatically analysed*, allowing thus not only to gather the information required to calculate PPI

* This work has been partially supported by the European Commission (FEDER), Spanish Government under the CICYT project SETI (TIN2009-07366); and projects THEOS (TIC-5906) and ISABEL (P07-TIC-2533) funded by the Andalusian local Government.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 125-128, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

values, but also to infer knowledge to answer questions like *what are the business process elements related to PPI P?* (a set of 3 analysis operation families amenable to be performed on PPI definitions can be found in [3]).

We address these issues by providing PPINOT tool, that allows the graphical definition of PPIs together with their corresponding BPs, and their subsequent automated analysis. To the best of our knowledge, there not exists any similar tool for the definition of PPIs. Concretely, we can highlight the following PPINOT features that give the novelty to our proposal:

BPMN 2.0 compliant. PPIs can be defined over BP diagrams (BPDs) previously modelled using the de facto standard BPMN 2.0.

Graphical definition of PPIs. PPINOT supports the graphical definition of PPIs using a graph-based graphical notation that is easily understandable by non-technical users, at the same time that it is supported by a metamodel that assures the precise and complete definition of PPIs.

Automated analysis of PPIs. The aforementioned metamodel support allows to automatically formalise PPI definitions using Description Logics, enabling to obtain information about the way PPIs and BP elements influence each other. Concretely, two kinds of analysis operations are supported in the current version of PPINOT: (I) *BPElements involved*, that allows to answer the question *Given a PPI P, Which are the process model's elements involved?*, this information is very useful in many scenarios, like for instance when a PPI must be replaced with others (maybe because it is very costly to obtain its value) and it is necessary to assure that every element of the BP that was measured before is measured in the new case; and (II) *PPIs associated to BPElement*, that allows to answer the question *Given a BPElement E, Which are the PPIs associated or applied to them?*, this information can assist during the evolution of BPs, e.g. if a part of the BP has evolved and is modified, for instance if an activity is deleted, this analysis allows to identify which PPIs will be affected and should be updated.

Figure 1 shows a screenshot of PPINOT tool in use.

2 PPINOT Tool: Definition and Structure

The structure of PPINOT is depicted in the component model of Figure 2. This tool has been implemented as an extension of the ORYX platform. Concretely we have provided a new stencil set with the shapes and connectors of the PPI graphical notation (PPINOT Oryx stencilset), that extends the existing one of BPMN. In addition, a new plugin called *PPINOT analyser* has been developed. It supports the formalisation of PPI graphical definitions to DL, and the subsequent analysis. Furthermore, it has been designed as a reusable component so that it can be integrated into other environments than Oryx without any change. In the following we describe the way PPINOT works.

A BPD is defined in *Oryx* [4]. Then, the set of PPIs is defined over such BPD using the *PPINOT Oryx Stencilset* . An xml file containing all this information (BPD + PPIs) is obtained from Oryx (through the *PPINOT Service*) and

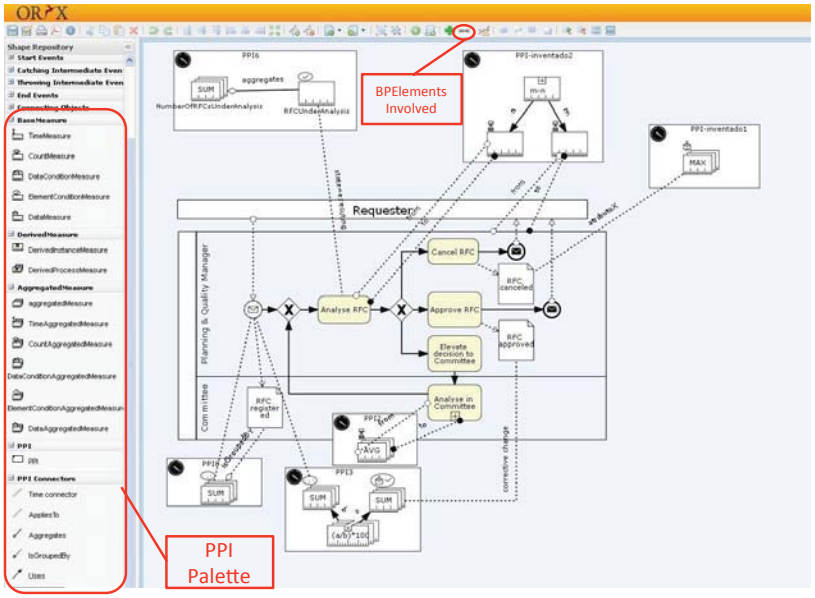


Fig. 1. PPINOT screenshot

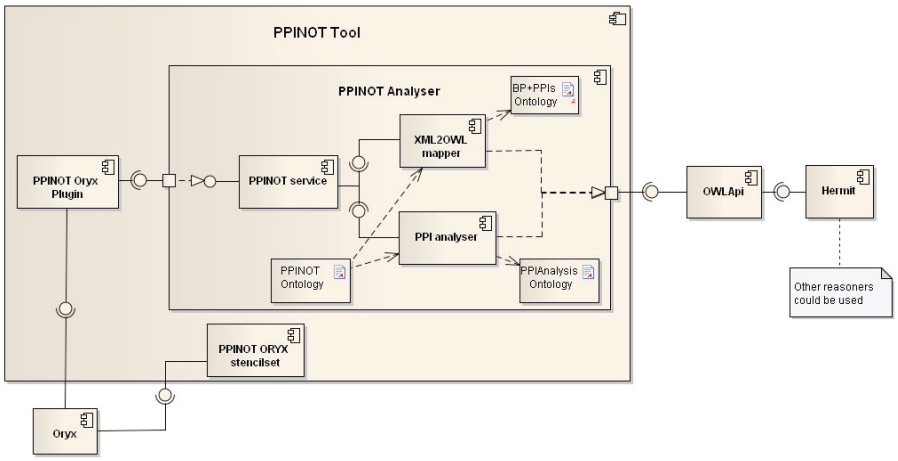


Fig. 2. PPINOT component model

mapped to OWL using the PPINOT ontology described in [2]. This is done by the *XML2OWL Mapper* and produces the OWL file BP+PPIs Ontology, which is the target file of the DL reasoner (in this case *HermiT*) used by the *PPI Analyser*, so the proper DL operations are executed on the PPI definitions of this OWL file to infer the information required. Finally, an OWL file containing the

information required (elements involved in a PPI definition or PPIs associated to a given BP element) is automatically generated (PPIAnalysisOntology).

3 PPINOT Structure

In this section we provide the steps we will follow in order to try PPINOT Tool.

1. In oder to access the tool a firefox window must be opened and the url `http://labs.isa.us.es:8081/backend/poem/repository` accessed.
2. There are several BPDs available or a new on can be created using the BPMN 2.0 stencil set.
3. Once the BP is open, the PPI extension must be added in order to be able to define the set of PPIs corresponding to that process.
4. Using the PPI palette, these PPIs must be modelled.
5. The following step is elated to the automated analysis of PPIs. In this case, we will try the *BPElements involved* operation. In order to try it, the *PPINOT query* plugin must be selected.
6. A measureDefinition from the list of all measureDefinitions shown by the system (corresponding to the PPIs defined) has to be selected.
7. The system provides a list with all the BP elements involved in the selected measureDefinition, and hence, involved in the corresponding PPI.

4 Conclusions

In this work we present PPINOT, an easy-to-use tool for the definition and automated analysis of PPIs. PPINOT satisfies the necessity of a tool that, on the one hand, fills the visual gap between BPs and their corresponding PPIs by allowing the modelling of such PPIs together with the corresponding BP; and on the other hand, automates the error-prone and tedious task of analyse PPIs. We plan to extend PPINOT in order to support the whole set of analysis operations identified in [3]. Another direction we are working on is to use PPINOT in order to obtain the PPIs' values from the execution of BPs in a BPMS.

References

1. Franceschini, F., Galetto, M., Maisano, D.: Management by Measurement: Designing Key Indicators and Performance Measurement Systems. Springer Verlag (2007)
2. del Río-Ortega, A., Resinas, M., Ruiz-Cortés, A.: Defining process performance indicators: An ontological approach. In: Proceedings of the 18th International Conference on Cooperative Information Systems (CoopIS). OTM 2010, Part I. (October, 2010) 555–572
3. del Río-Ortega, A., Resinas, M., Ruiz-Cortés, A.: PPI definition and automated design-time analysis. Technical report, Applied Software Engineering Research Group (2012)
4. Decker, G., Overdick, H., Weske, M.: Oryx - an open modeling platform for the bpm community. In: BPM. (2008) 382–385

Sesión 4

Ingeniería de Servicios II

Chair: *Dr. Vicente Pelechano*

Sesión 4: Ingeniería de Servicios II
Chair: Dr. Vicente Pelechano

Jenifer Verde, Juan Manuel Vara, Veronica Andrea Bollati and Esperanza Marcos. *Desarrollo de puentes tecnológicos para soportar el modelado de interfaces de servicio*.

Rubén Casado, Javier Tuya and Muhammad Younas. *An Abstract Transaction Model for Testing the Web Services Transactions*.

José María García, David Ruiz, and Antonio Ruiz-Cortés. *A Model of User Preferences for Semantic Services Discovery and Ranking*.

M.Carmen De Castro, Azahara Camacho-Magriñán and Inmaculada Medina-Bulo. *Aplicación de la técnica de las pruebas metamórficas a una composición de servicios: Metasearch*.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): Actas de las "VIII Jornadas de Ciencia e Ingeniería de Servicios (JCIS'2012)", Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

Desarrollo de puentes tecnológicos para soportar el modelado de interfaces de servicio

Jenifer Verde, Juan M. Vara, Veronica Bollati y Esperanza Marcos

Grupo de Investigación Kybele, Universidad Rey Juan Carlos, Madrid, España
{jenifer.verde, juanmanuel.vara, veronica.bollati, esperanza.marcos}@urjc.es

Resumen. Este trabajo presenta el desarrollo de puentes tecnológicos que permiten extraer modelos de Descripciones Abstractas de Servicios a partir de especificaciones WSDL existentes y viceversa. Para ello, se presentan un conjunto de DSLs que se utilizan para la elaboración de algunos modelos intermedios durante el proceso de extracción y las transformaciones de modelos que los conectan, automatizando el proceso. Los modelos obtenidos permiten implementar cualquier proceso de razonamiento acerca de la interfaz de uno o varios servicios utilizando técnicas propias de la Ingeniería Dirigida por Modelos, como transformaciones, validadores, etc. Así, este trabajo proporciona una base tecnológica sobre la que abordar nuevas propuestas metodológicas en el futuro.

Palabras clave: WSDL; Interfaz de Servicio; Ingeniería Dirigida por Modelos; Transformaciones de Modelos; DSL.

1 Introducción

Cualquier sistema software está sometido a la constante necesidad de evolucionar para responder a los avances tecnológicos y a los cambios que se producen en la lógica de negocio de dichos sistemas. En particular, la evolución es un aspecto crítico en el área de la orientación a servicios, pues resulta muy complicado introducir cambios en los sistemas distribuidos y heterogéneos que resultan de aplicar el paradigma de desarrollo orientado a servicios [14].

Por otro lado, una de las aproximaciones que durante los últimos años ha cobrado mayor importancia en la Ingeniería del Software es la Ingeniería Dirigida por Modelos (*Model-Driven Engineering*, MDE) [16]. Sus principales características pasan por potenciar el papel de los modelos y el nivel de automatización en cualquier actividad relacionada con el desarrollo de software [2].

Combinando estas dos ideas, en trabajos anteriores presentamos una primera aproximación a un marco tecnológico dirigido por modelos para soportar la evolución de servicios a nivel de interfaz [7], entendida en este contexto como la posibilidad de reemplazar un servicio por otra versión de ese mismo servicio sin *romper* a sus consumidores. Para ello, se desarrolló un Lenguaje Específico de Dominio (*Domain-Specific Language*, DSL) [13] para modelar descripciones abstractas de servicio (*Abstract Service Description*, ASD) y se construyó un comparador básico, que implementaba la noción de compatibilidad de servicios descrita formalmente en [1], para evaluar la compatibilidad de dos versiones de un servicio descritas con dicho

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 131-144, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

DSL. No obstante, aún cuando existen otras iniciativas para la representación de servicios [5, 20], la mayor parte de los servicios existentes optan por utilizar WSDL [21, 22] como notación para proporcionar una descripción estándar de su interfaz.

Así, este trabajo aborda una de las líneas de trabajo futuro identificadas en [7], presentando el desarrollo de un conjunto de puentes tecnológicos en forma de extractores e inyectores para poder extraer la información recogida en la especificación WSDL de un servicio y expresarla en términos del DSL para el modelado de ASDs. De esta forma, se pueden obtener modelos ASD a partir de la descripción WSDL de cualquier servicio. Como resultado, podemos elevar el nivel de abstracción al que razonamos sobre la interfaz de uno o varios servicios y automatizar cualquier proceso de razonamiento sobre dichas interfaces, utilizando técnicas MDE. Por ejemplo, podríamos construir generadores de servicios que, a partir de la interfaz de dos o más servicios produjese un nuevo servicio que aunase la funcionalidad de todos ellos, utilizando técnicas de *model merging* [3]; realizar verificaciones formales sobre las propiedades de la interfaz utilizando técnicas de *model checking* [6] o soportar la comparación de versiones como hacemos en [7].

Igualmente, los puentes tecnológicos que se presentan en este trabajo soportan el camino inverso. Es decir, permiten generar documentos WSDL a partir de descripciones abstractas de servicio en forma de modelos ASD. De esta forma, el resultado de implementar razonadores con técnicas MDE podría expresarse en forma de nuevos modelos ASD que podrían utilizarse para generar la descripción WSDL de los servicios modelados.

En la literatura, los procesos soportados por estos puentes tecnológicos son normalmente conocidos como inyección (construir un modelo a partir de un fichero de código) y extracción de modelos (generar un fichero de código a partir de un modelo) [4]. Por motivos de espacio en este trabajo nos centraremos en presentar el proceso de inyección.

El resto del artículo se estructura como sigue: la sección 2 presenta una breve descripción del proceso de desarrollo seguido, los distintos DSLs y las transformaciones que los conectan. La sección 3 utiliza un caso de estudio para ilustrar el funcionamiento de la propuesta y finalmente la sección 4 resume las principales conclusiones e identificando algunas líneas para trabajos futuros.

2 Desarrollo de puentes tecnológicos

Esta sección presenta el desarrollo de los puentes tecnológicos que permiten obtener modelos ASD a partir de la especificación WSDL de un servicio Web. Para ello, se describe primero el proceso de inyección y a continuación los artefactos que lo soportan, básicamente DSLs para la elaboración de modelos intermedios y transformaciones de modelos que conectan dichos DSLs.

2.1 Descripción del proceso

A continuación se proporciona una vista general del proceso de inyección de modelos soportado por los puentes tecnológicos presentados en este trabajo.

El proceso de inyección de modelos, que se muestra en la Fig. 1, está compuesto por una serie de pasos. Cada uno de estos pasos implica una transformación de

modelos [17]. En particular, la primera es una transformación de texto a modelo (t2m), mientras que las siguientes son transformaciones de modelo a modelo (m2m). A continuación se describe brevemente cada paso:

- El punto de partida es la especificación WSDL que describe la interfaz pública de un servicio Web (S). Dicha especificación no es más que un documento XML, conforme a un XML Schema (`WSDL.xsd`). Este fichero es tomado como entrada por una primera transformación t2m que convierte la especificación WSDL de un servicio en un modelo XML conforme a un metamodelo XML (`XML.ecore`). Por lo tanto, esta transformación permite elevar el nivel de abstracción, pasando del mundo de las gramáticas al mundo de los modelos [23]. A partir de este momento, podríamos aplicar cualquiera de las técnicas de procesamiento de modelos más comunes en MDE, como transformaciones, *weaving* ó *merging* [2].
- A continuación, una transformación m2m (`XML2WSDL`) toma como entrada el modelo XML anterior, donde encontramos objetos de tipo `element`, `attribute`, etc. y genera un modelo WSDL, compuesto por objetos `operation`, `message`, etc. Este modelo es obviamente conforme a un metamodelo WSDL (`WSDL.ecore`).
- Finalmente, el modelo WSDL obtenido es consumido por una nueva transformación (`WSDL2ASD`) que devuelve un modelo ASD conforme al metamodelo ASD (`ASD.ecore`). Dicho modelo contiene la descripción abstracta del servicio en términos del DSL para el modelado de ASDs presentado en [7]. En esencia, proporciona una descripción de la interfaz de alto nivel, que es independiente de la tecnología concreta con que se implemente.

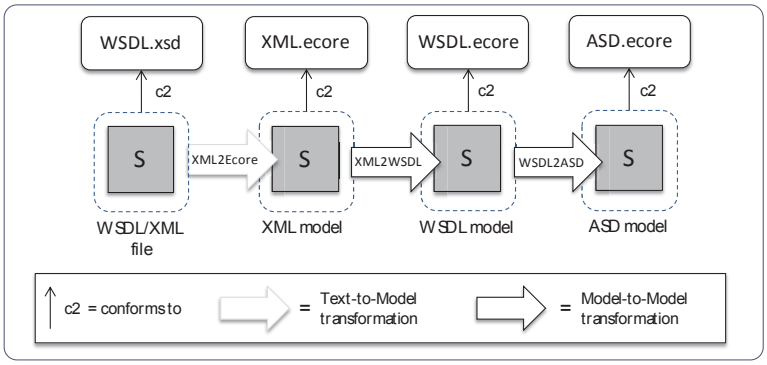


Fig. 1. Proceso de desarrollo de puentes tecnológicos

Conviene aclarar la necesidad del paso intermedio por un modelo XML. En efecto, podríamos tratar de extraer directamente un modelo WSDL a partir del fichero WSDL, pero en tal caso necesitaríamos codificar un parser ad-hoc que recorriese el fichero WSDL generando un nuevo documento XML que sería el modelo WSDL (los modelos que manejamos en este trabajo se persisten en formato XML). No obstante, esta tarea, además de tediosa y laboriosa, resultaría útil sólo en este escenario. Es decir, sólo nos permitiría extraer modelos de ficheros WSDL. Si quisiéramos aplicar la misma técnica con cualquier otro tipo de fichero XML, como por ejemplo un fichero BPEL o SOAP, tendríamos que desarrollar nuevos parsers.

En cambio, la aproximación aplicada en este trabajo, que se basa en la generación de un modelo XML intermedio, es extrapolable a cualquier otro proceso de inyección de modelos a partir de ficheros XML. Una vez obtenido el modelo XML a partir del correspondiente fichero BPEL, sólo habría que transformarlo a un modelo BPEL conforme a un metamodelo BPEL. Para ello, se codificaría una transformación m2m, que, dentro de la complejidad inherente al desarrollo de transformaciones, resulta menos compleja que una transformación t2m (desarrollo de un nuevo parser). De hecho, esta aproximación es la que se aplica habitualmente en escenarios que implican acercar diferentes espacios tecnológicos [12, 15].

Conviene mencionar que en la Web de ATL existe una transformación XML2WSDL. No obstante, el metamodelo que incluye es una versión reducida de WSDL 2.0. Por ejemplo, dicho metamodelo no soporta el modelado de espacios de nombres, por lo que fue necesario definir un metamodelo más completo que solventara estos problemas y por tanto nuevas transformaciones. Además, hubo de definirse un nuevo metamodelo para la versión 1.1 del estándar. Para ello se utilizo como base la DTD propuesta por la W3C. Por último, en ambas transformaciones XML2WSDL el metamodelo XML utilizado es el que se encuentra en la Web de ATL.

2.2 Definición de DSLs intermedios

Esta sección presenta los diferentes DSLs que se utilizan para elaborar los modelos que se producen en los distintos pasos del proceso presentado en la sección anterior.

Para definir la sintaxis abstracta de estos DSLs se implementan sus metamodelos con Ecore, el lenguaje de metamodelado de EMF [18].

Respecto a la sintaxis concreta (nos limitamos a representaciones visuales y no textuales), a partir de un metamodelo Ecore, EMF permite generar un editor básico de tipo árbol que permite editar modelos conformes a dicho metamodelo. Estos editores genéricos han sido personalizados para mejorar su usabilidad y que resulten más intuitivos.

Además, se han utilizado las facilidades proporcionadas por GMF [8] y Eugenia [10] para mejorar el editor gráfico desarrollado para el modelado de ASDs presentado en [7]. En particular, se ha utilizado el lenguaje EOL (*Epsilon Object Language*) [11], un lenguaje imperativo para implementar operaciones de gestión de modelos, para codificar un fichero que recoge las decisiones de diseño respecto al editor. Por ejemplo, dicho fichero especifica los detalles de la representación gráfica deseada para cada elemento del metamodelo. De esta forma se evita contaminar el metamodelo con anotaciones para recoger estas decisiones de diseño. Además, permite generar diferentes representaciones para un mismo metamodelo sin necesidad de cambiar el metamodelo sin más que construir distintos ficheros EOL que especifiquen distintas opciones de representación.

Dado que el proceso de construcción de editores puede consultarse en las referencias existentes [8, 18], a continuación se describen brevemente los diferentes metamodelos que definen la sintaxis abstracta de cada DSL utilizado en el proceso de inyección de modelos que se presenta en este trabajo, de acuerdo al orden que se presentan en la Fig. 1.

2.2.1. Modelado de descripciones abstractas de servicios (ASD)

En la Fig. 2 se muestra parcialmente el metamodelo ASD que contiene las siguientes metaclases:

- `ServiceDescription`: elemento raíz del modelo que contiene el resto de elementos.
- `Operation`: elemento que representa las operaciones que ofrece el servicio.
- `Message`: mensajes que se deben intercambiar para llevar a cabo una operación.
- `InfoType` e `InfoTypeImported`: representan los tipos de datos XSD simples o compuestos que se utilizan dentro de los mensajes.
- `Profile`: representa un conjunto de `AssertionSets` que determinan un perfil concreto que se aplica para definir un cierto nivel de calidad para un servicio.
- `AssertionSet`: agrupa un conjunto de `Assertions`.
- `Assertion`: representa las condiciones o requisitos que tiene que cumplir un servicio.

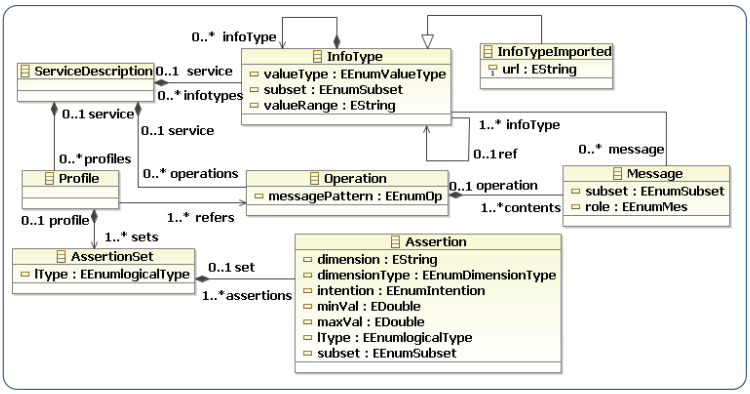


Fig. 2. Metamodelo ASD

2.2.2. Modelado de especificaciones WSDL (1.1)

Para soportar el modelado de documentos WSDL conforme a las dos versiones del estándar, se han desarrollado dos DSLs diferentes, uno para la versión 1.1 [21] y otro para la versión 2.0 [22], que presentan considerables diferencias.

La decisión de soportar las dos versiones del estándar surge de la necesidad de disponer de ejemplos de servicios Web funcionando en entornos reales. Aunque la versión WSDL 2.0 es la que está vigente actualmente, la mayor parte de los servicios encontrados utilizan WSDL 1.1 para describir su interfaz, probablemente porque los desarrolladores prefieran utilizar la versión anterior del estándar, más consolidada y estable.

En la Fig. 3 se puede ver una versión simplificada del metamodelo para la versión 1.1 del estándar, que recoge sólo las metaclases más destacadas (y las relaciones entre ellas):

- `Definitions`: es el elemento raíz de la especificación WSDL 1.1.
- `Message`: representa los mensajes que se intercambian entre el cliente y el servicio para realizar una determinada operación.

- `Part`: identifica cada una de las partes en las que se divide un mensaje.
- `PortType`: representa la interfaz que contiene todas las operaciones que ofrece el servicio (conjunto de `portTypeOperation`). Dentro de un elemento `portTypeOperation` se pueden definir distintos elementos que dependen del tipo de mensaje que se intercambie para llevar a cabo una operación concreta (`PortTypeFault` contiene un mensaje de notificación error, `PortTypeInput` contiene un mensaje de entrada y `PortTypeOutput` contiene un mensaje de salida).
- `Service`: representa el conjunto de puertos (`Port`) a los que está atado un servicio.
- `Binding`: es un conjunto de `BindingOperation`. Cada `BindingOperation` aporta la información necesaria sobre cómo debe ser el formato de los mensajes y el protocolo que se va a utilizar para intercambiar dichos mensajes que permiten realizar una determinada operación definida dentro del elemento `portType`. Cada `BindingOperation` está asociado a un `PortTypeOperation`. Dependiendo del tipo de operación a la que vaya asociado se utilizarán los siguientes elementos: `BindingInput` (asociado a `PortTypeInput`), `BindingOutput` (asociado a `PortTypeOutput`) y `BindingFault` (asociado a `PortTypeFault`).
- `ElementType (Types)`: es uno de los elementos más complejos del metamodelo ya que dentro de él se definen los tipos de datos que se van a utilizar en los mensajes. Tanto en la versión 1.1 de WSDL como en la versión 2.0 se apoya en la especificación de XML Schema siguiendo su DTD [24, 25] para la definición de los tipos de datos.
- `Import`: representa los tipos de datos que se importan de otros XML Schema.
- `Namespace`: aunque en la especificación de WSDL no existe este elemento se ha añadido para poder soportar la utilización de espacios de nombres.

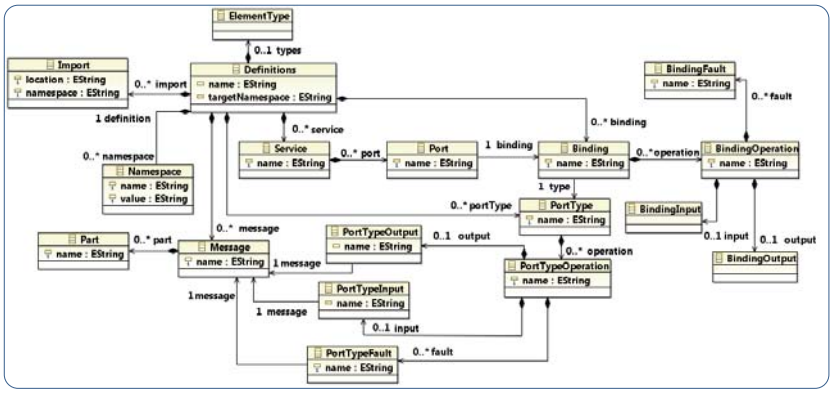


Fig. 3. Metamodelo WSDL 1.1

Aparte de los elementos pertenecientes al XML Schema que se utilizan en el metamodelo WSDL 1.1, este también incluye elementos que lo extienden pertenecientes a MIME, SOAP 1.1 y HTTP 1.1, aunque no se muestran en la Fig. 3.

2.2.3. Modelado de especificaciones WSDL (2.0)

Como hemos mencionado, WSDL 2.0, la última versión del estándar propuesto por la W3C, presenta ciertas diferencias con respecto a la versión anterior, WSDL 1.1.

En particular, la Tabla 1 resume las principales diferencias en cuanto al renombrado de conceptos.

WSDL 1.1	WSDL 2.0
Elemento `Definitions` (nodo raíz)	Elemento `Description` (nodo raíz)
Elemento `PortType`	Elemento `Interface`
Atributo `type` del elemento `Binding`	Atributo `interface` del elemento `Binding`
Elemento `Port`	Elemento `EndPoint`

Tabla 1. Diferencias entre el estándar WSDL 1.1 y WSDL 2.0

Además, existen diferencias más relevantes en cuanto a la obligatoriedad de ciertos atributos, la cardinalidad de algunas relaciones, etc. A continuación se resumen algunas de las más importantes:

- El atributo `targetNamespace` del elemento `description` pasa a ser obligatorio.
- Al nodo raíz (`description`) se le añade el elemento `includes`, para poder considerar en el documento tipos de datos definidos en otros documentos.
- El elemento `message` también desaparece y los mensajes pasan a definirse dentro del elemento `interface`.
- La cardinalidad del elemento `endPoint` que es el equivalente al elemento `port` pasa a tener cardinalidad 1 a N.

Tal y como se puede comprobar, los cambios entre una versión del estándar y otra son numerosos y relevantes, lo que nos llevó a desarrollar un DSL para el modelado de documentos WSDL conformes a la nueva versión del estándar. Adicionalmente hubo que desarrollar dos versiones de las transformaciones `XML2WSDL y WSDL2ASD`.

Finalmente, la Fig. 4 muestra una versión simplificada del metamodelo definido para el modelado de documentos WSDL 2.0. Nótese que es sólo una vista parcial, ya que el metamodelo final resulta muy complejo porque incluye la implementación de los elementos de extensibilidad y los tipos de datos propios de XML Schema, como `complexType`, `group`, etc., que necesitamos a la hora de definir un modelo WSDL.

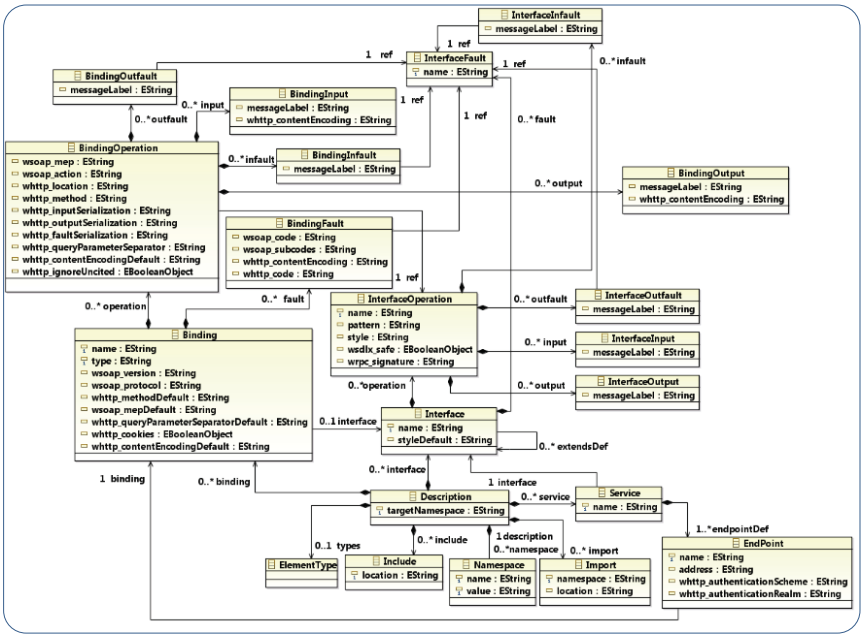


Fig. 4. Metamodelo WSDL 2.0

2.3 Transformaciones de modelos

Como se ha mencionado en la descripción del proceso (sección 2.1), para obtener un modelo ASD a partir de una especificación WSDL se necesitan tres transformaciones sucesivas: una primera transformación t2m que pasa de fichero XML a modelo XML; una segunda transformación m2m que genera un modelo WSDL a partir del modelo XML y una última transformación m2m que produce un modelo ASD a partir de un modelo WSDL.

Para la primera transformación se ha utilizado el inyector/extractor de modelos que proporciona el IDE de ATL, basado en un parser SAX. Para poder utilizar tanto el inyector como el extractor sólo fue necesario desarrollar nuevos lanzadores, ya que las últimas versiones del IDE no los incluían. Por lo tanto, a continuación nos centraremos en presentar las dos transformaciones m2m. Nótese que son en realidad cuatro transformaciones ya que consideramos los dos DSLs para las dos versiones del estándar, aunque por motivos de espacio presentaremos sólo las de la versión 1.1.

Para desarrollar estas transformaciones, primero se especifican a alto nivel las reglas de transformación, identificando las correspondencias entre los elementos de los modelos de entrada y los elementos de los modelos de salida y a continuación se implementan utilizando el lenguaje de transformación ATL [9]. Con el fin de ilustrar la funcionalidad soportada por estas transformaciones, a continuación se resumen las correspondencias entre los elementos de los diferentes modelos.

2.3.1. De modelos XML a modelos WSDL

Aunque las correspondencias que implementa la transformación entre modelos XML y modelos WSDL son muy intuitivas, la tarea de codificación resulta bastante tediosa por la cantidad de elementos distintos que contiene el metamodelo WSDL.

Para ilustrarla, la siguiente tabla muestra las relaciones de correspondencia entre los elementos más importantes del metamodelo XML y el metamodelo para WSDL 1.1. Nótese que son similares a las que existen con el metamodelo para WSDL 2.0, considerando las diferencias que existen entre las dos versiones del estándar.

XML	WSDL 1.1
`Root`	`Definitions`
`Element` con atributo `name=Types`	`ElementType`
`Element` con atributo `name=Message`	`Message`
`Element` con atributo `name=Part`	`Part`
`Element` con atributo `name=PortType`	`PortType`
`Element` con atributo `name=Binding`	`Binding`
`Element` con atributo `name=Service`	`Service`
`Element` con atributo `name=Port`	`Port`
`Element` con atributo `name=Operation` y está contenido dentro de `Element` con atributo `name=PortType`	`PortTypeOperation`
`Element` con atributo `name=Input` y está contenido dentro de `Element` con atributo `name=PortType`	`PortTypeInput`
`Element` con atributo `name=Output` y está contenido dentro de `Element` con atributo `name=PortType`	`PortTypeOutput`
`Element` con atributo `name=Fault` y está contenido dentro de `Element` con atributo `name=PortType`	`PortTypeFault`
`Element` con atributo `name=Operation` y está contenido dentro de `Element` con atributo `name=Binding`	`BindingOperation`
`Element` con atributo `name=Input` y está contenido dentro de `Element` con atributo `name=Binding`	`BindingInput`
`Element` con atributo `name=Output` y está contenido dentro de `Element` con atributo `name=Binding`	`BindingOutput`
`Element` con atributo `name=Fault` y está contenido dentro de `Element` con atributo `name=Binding`	`BindingFault`
`Element` con atributo `name=Import` y está contenido dentro de `Root`	`Import`
`Attribute` con atributo `name` que comience por `xmlns`	`Namespace` con atributo `name` igual al atributo `name` del elemento `Attribute`
`Attribute` con atributo `name` y `value`	Atributo dentro de un elemento cuyo nombre es igual al atributo `name` y cuyo valor es igual al atributo `value`

Tabla 2. Correspondencias entre los elementos del metamodelo XML y los elementos del metamodelo para WSDL 1.1

2.3.2. De modelos WSDL a modelos ASD

Por último, la Tabla 3 resume las relaciones de correspondencia entre los principales elementos del metamodelo WSDL 1.1 y el metamodelo ASD. De nuevo, nótese que para identificar las relaciones entre los elementos del metamodelo WSDL 2.0 y el metamodelo ASD, partimos de esta tabla y la combinamos con los cambios que existen entre las dos versiones del metamodelo para WSDL.

WSDL 1.1	ASD
`Definitions`	`ServiceDescription`
Atributo `name` de `Definitions`	Atributo `name` de `ServiceDescription`
`PortTypeOperation`	`Operation`
`PortTypeInput`, `PortTypeOutput`, `PortTypeFault`	`Message`
`PortTypeInput` → `role=input`, `PortTypeOutput` → `role=output`, `PortTypeFault` → `role=fault`	Atributo `role` de `Message`
`PortTypeInput` → `subset=req`, `PortTypeOutput` y `PortTypeFault` → `subset=pro`	Atributo `subset` de `Message`
Número y orden de los elementos `PortTypeInput`, `PortTypeOutput` y `PortTypeFault` que contiene el elemento `PortTypeOperation`	Atributo `messagePattern` del elemento `Operation`
`Element` , `ComplexType` (sólo si son hijos de XML Schema) y elementos `SimpleType` (sólo cuando es un tipo primitivo)	Elemento `InfoType`
Atributo `primitivetype` del elemento `Element`	Atributo `valueType` del elemento `InfoType`
Atributo `minOccurs` y `maxOccurs` del elemento `Element`	Atributo `valueRange` del elemento `InfoType`

Tabla 3. Correspondencia entre los elementos del metamodelo para WSDL 1.1 y los elementos del metamodelo para el modelado de ASDs

3 Caso de Estudio

Para ilustrar el funcionamiento de los puentes tecnológicos construidos en este trabajo, hemos utilizado varios casos de estudio. En esta sección se presenta uno de ellos, tomado de la *S-Cube Network of Excellence*[1], en particular el escenario *Automotive Purchase Order Processing*, desarrollado y utilizado como uno de los escenarios de validación en dicha red. El escenario está basado en el modelo llamado *Supply Chain Operations Reference* (SCOR), que proporciona directrices para la implementación de cadenas de suministro. El escenario en cuestión, que es parte de los procesos de una compañía de la industria de la automoción, es un ejemplo de cómo implementar actividades de nivel 3 de SCOR utilizando SOA.

En particular, mostramos la obtención de un modelo ASD a partir de la especificación WSDL de un servicio para la adquisición on-line de automóviles. Para ello, en primer lugar se genera un modelo utilizando el inyector de XML que incluían versiones anteriores del IDE ATL. La Fig. 5 ilustra de forma parcial este primer paso.

[1] http://www.s-cube-network.eu/

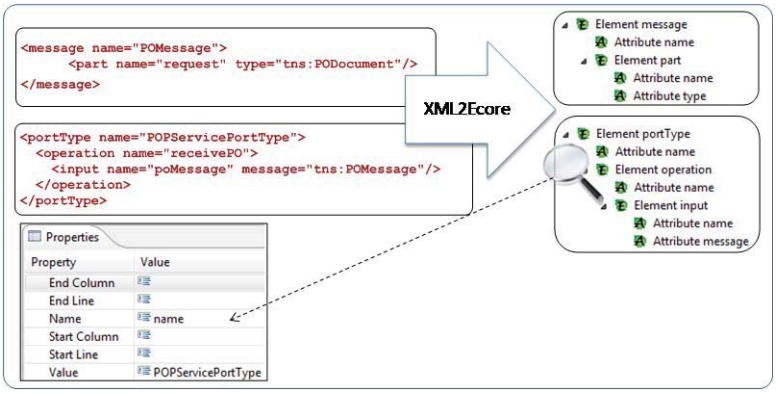


Fig. 5. Generación de un modelo XML a partir de un fichero WSDL

A continuación, el modelo XML generado es utilizado para producir un modelo WSDL conforme al metamodelo WSDL 1.1 descrito en la Sección 2.2.2.

La parte superior de la Fig. 6 da una vista general del resultado de dicha transformación.

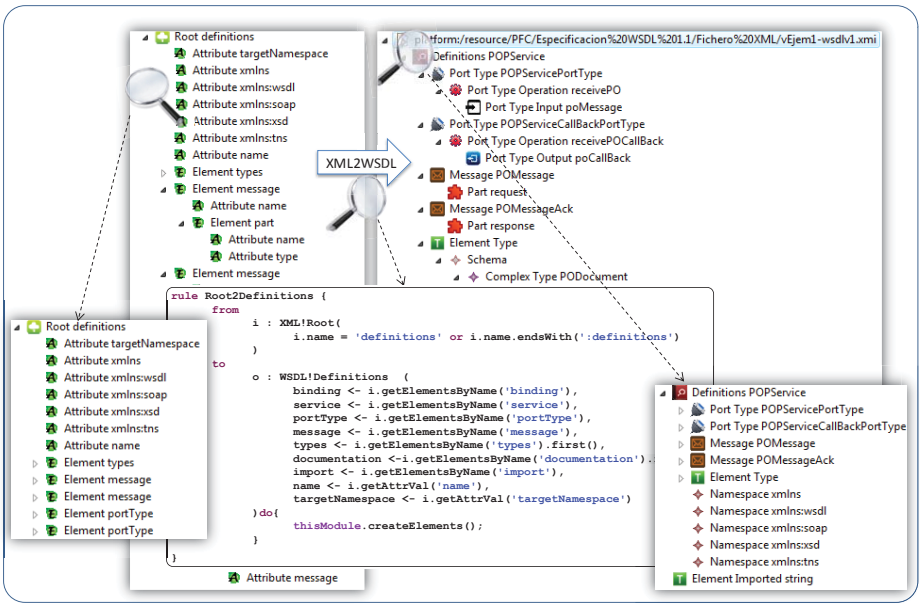


Fig. 6. Transformación `XML2WSDL`: regla `Root2Definitions`

. En la parte inferior se muestra a modo de ejemplo una regla sencilla (`Root2Definitions`), que transforma el elemento raíz del modelo XML (modelo origen) en el elemento `Definitions` del modelo WSDL (modelo destino). Este elemento puede contener a su vez varios elementos como mensajes (`message`), conjuntos de operaciones (`portType`) o de puertos (`service`), tipos de datos (`types`), comentarios (`document`), `bindings` y otros tipos de datos pertenecientes a otros XML Schemas (`import`). Para recuperarlos se utiliza la función auxiliar `getElementsByname()` que obtiene el conjunto de elementos de tipo `Element` del

modelo XML de entrada cuyo nombre coincide con el que se pase como argumento. El motor de transformación se encarga entonces de identificar cuáles son los objetos del modelo destino a los que dan lugar los elementos origen recuperados.
De forma análoga, la función auxiliar `getAttrVal()` permite definir los atributos del elemento `Definitions`. Por último, la función `createElements()` invocada de manera imperativa al final de la regla, se encarga de mapear correctamente la información referida a los espacios de nombres XML que habían sido definidos en el documento WSDL original. Para ello identifica los atributos del elemento `Root` del modelo XML cuyo nombre contiene el prefijo `xmlns` y por cada uno añade un objeto `Namespace` al modelo WSDL destino.

Finalmente, el último paso para conseguir el objetivo propuesto es convertir el modelo WSDL 1.1 resultante en un modelo ASD. Esto se consigue mediante otra transformación ATL (`WSDL2ASD`) que en cierto modo filtra la información del modelo WSDL para quedarse sólo con la parte relevante desde el punto de vista de obtener una descripción de la interfaz de alto nivel de abstracción.

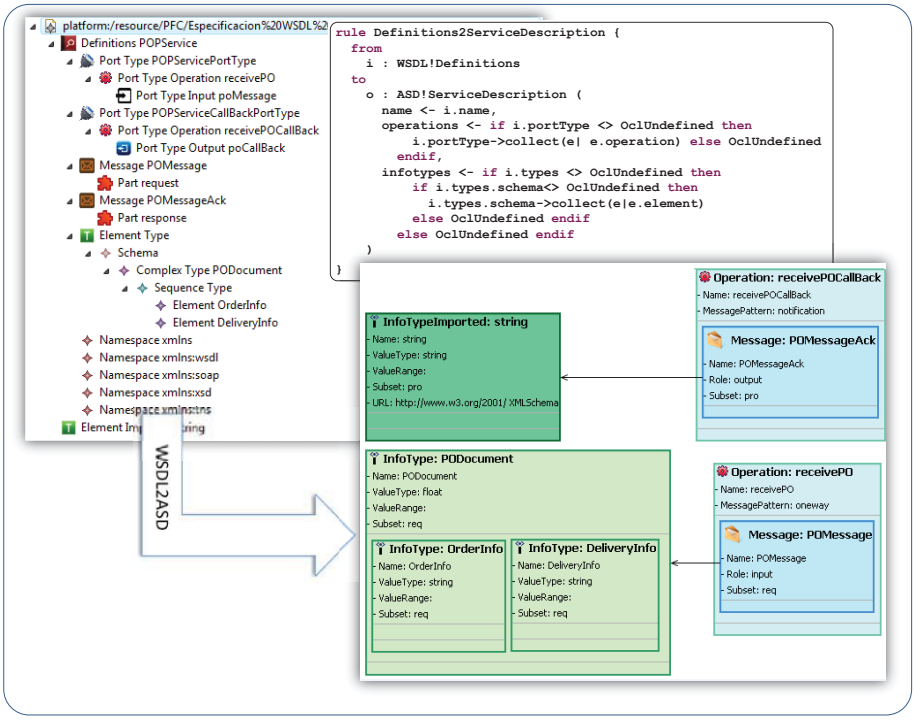


Fig. 7. Transformación `WSDL2ASD`: regla `Definitions2ServiceDescription`

La Fig. 7 proporciona una vista general del modelo de salida (en el editor GMF [19] mejorado) y la regla `Definitions2ServiceDescription` de la transformación. Dicha regla produce un elemento `ServiceDescription` a partir del elemento `Definitions` del modelo WSDL de entrada. Además, los elementos destino `InfoType` y `Operations` que serán creados por otras reglas para mapear los elementos origen `ElementType` y `PortType`, son asignados al nuevo objeto `ServiceDescription`.

4 Conclusiones y trabajos futuros

Este trabajo continúa la línea iniciada en [7] centrada en el desarrollo de un marco para soportar la evolución de servicios. La idea subyacente es aprovechar las ventajas proporcionadas por la MDE en términos de automatización y nivel de abstracción [16], para dar soporte a propuestas metodológicas basadas en la orientación a servicios.

En particular, se ha presentado el desarrollo de los puentes tecnológicos que permiten obtener una descripción de alto nivel de la interfaz de un servicio Web a partir de su especificación WSDL, utilizando para ello un conjunto de DSLs conectados por transformaciones de modelos.

Aunque originalmente el objetivo de este trabajo era sólo dar un paso más hacia la construcción del marco mencionado, en realidad los resultados obtenidos ofrecen una base tecnológica sobre la que construir nuevas propuestas. Como mencionábamos en la introducción, una vez que disponemos de una descripción de alto nivel del servicio podemos plantearnos otros objetivos además de soportar la comparación. Por ejemplo, en la actualidad trabajamos para generar una nueva versión compatible a partir de la descripción de dos o más versiones de un mismo servicio que no lo sean. Para ello producimos un nuevo modelo ASD que es utilizado para generar un nuevo documento WSDL utilizando los extractores que constituyen la otra parte de los puentes tecnológicos que, por motivos de espacio, no hemos presentado en este trabajo.

Igualmente, se plantea la definición de un único metamodelo para las dos versiones del estándar y la correspondiente unificación de las distintas transformaciones, así como una serie de mejoras sobre los artefactos software desarrollados. Por ejemplo, se están definiendo mecanismos de validación para asegurar que los modelos elaborados con los distintos DSLs pueden ser consumidos por las distintas transformaciones. Para ello, utilizamos el lenguaje EVL, basado en EOL [11] para implementar restricciones a nivel de metamodelo que son luego comprobadas a nivel de modelo.

Agradecimientos

Esta investigación ha sido llevada a cabo en el marco de trabajo del proyecto MASAI (TIN-2011-22617), financiado por el Ministerio de Ciencia e Innovación de España.

Referencias

1. Andrikopoulos, V. (2010). *A Theory and Model for the Evolution of Software Services*. Tilburg: Tilburg University Press. Available: http://arno.uvt.nl/show.cgi?fid=107815
2. Bernstein, P A. (2003). Applying Model Management to Classical Meta Data Problems. In First Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA.
3. Brunet, G., Chechik, M., Easterbrook, S., Nejati, S., Niu, N., & Sabetzadeh, M. (2006). A manifesto for model merging. *Intl Workshop on Global integrated model management GaMMa*, *06*, 5. ACM Press.
4. Canovas Izquierdo, J. L., Jouault, F., Cabot, J., & Molina, J. G. (2011). API2MoL: Automating the building of bridges between APIs and Model-Driven Engineering. *Information and Software Technology*, *54*(3), pp. 257-273.
5. CBDI-SAE, *CBDI-SAETMmeta model for SOA version 2.0*, Everware-CBDI Inc, 2007. [Online]. Available: http://www.cbdiforum.com/public/meta model v2.php

6. Combemale, B., Crégut, X., Garoche, P.-L., & Thirioux, X. (2009). Essay on Semantics Definition in MDE. An Instrumented Approach for Model Verification.*Journal of Software*, *4*(9), 943-958.
7. Granada, D., Vara, J.M., Andrikopoulos, V., Marcos, E. Aplicando la Ingeniería Dirigida por Modelos para soportar la evolución de servicios. Novática, Vol.214 (Noviembre-Diciembre 2011), pp. 52-56.
8. Gronback, R. C. (2009). *Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit*: Addison-Wesley Professional.
9. Jouault, F., Allilaire, F., Bézivin, J., & Kurtev, I. (2008), "ATL: A model transformation tool", Science of Computer Programming, vol. 72 (1-2), pp. 31-39.
10. Kolovos, D. S., Rose, L. M., Paige, R. F., & Polack, F. (2009). Raising the level of abstraction in the development of GMF-based graphical model editors. In Proceedings of the 2009 ICSE Workshop on Modeling in Software Engineering (pp. 13-19). Vancouver, Canada: IEEE Computer Society.
11. Kolovos, D., Paige, R., & Polack, F. (2006), "The Epsilon Object Language (EOL)". Model Driven Architecture – Foundations and Applications. vol. 4066, pp. 128-142, Springer.
12. Kurtev, I., Bezivin, J., & Aksit, M. (2002). *Technological Spaces: An Initial Appraisal*. Paper presented at the Confederated International Conferences DOA, CoopIS and ODBASE 2002, Industrial Track, Irvine, California, USA.
13. Mernik, M., Heering, J., & Sloane, A. M. (2005). When and how to develop domain-specific languages. *ACM Computer Surveys*, *37*(4), 316-344.
14. Papazoglou, M., & van den Heuvel, W.-J. (2007). Service oriented architectures: approaches, technologies and research issues. *The VLDB Journal The International Journal on Very Large Data Bases*, *16*(3), 389-415.
15. Parreiras, F. S., Staab, S., & Winter, A. (2007). On marrying ontological and metamodeling technical spaces. *Joint meeting of the European Software Eng Conf and the ACM SIGSOFT Symp on The Foundations of Software Eng ESECFSE* (p. 439). ACM Press.
16. Schmidt, D. C. (2006), "Model-driven engineering", IEEE Computer, vol. 39 (2), pp. 25-31.
17. Sendall, S., & Kozaczynski, W. (2003). Model Transformation–the Heart and Soul of Model-Driven Software Development. IEEE Software, 20(5), 42-45.
18. Steinberg, D., Budinsky, F., Paternostro, M., & Merks, E. (2008), EMF: Eclipse Modeling Framework, 2nd Edition ed.: Addison-Wesley Professional.
19. Tikhomirov, A., & Shatali, A. (2008). Introduction to the Graphical Modeling Framework. Tutorial at the EclipseCON 2008. Santa Clara, California.
20. Walkerdine, J. Hutchinson, J. Sawyer, P. Dobson, G. and Onditi, V. *A faceted approach to service specification*, in Proceedings of the Second International Conference on Internet and Web Applications and Services. IEEE Computer Society, 2007, p. 20.
21. Web Services Description Language (WSDL) 1.1: http://www.w3.org/TR/wsdl
22. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language: http://www.w3.org/TR/wsdl20/
23. Wimmer, M. Kramler, G. *Bridging Grammarware and Modelware*; 4th Workshop in Software Model Engineering (WiSME 2005), Montego Bay, Jamaica; 03.10.2005; in: *Satellite Events at the MoDELS 2005 Conference, LNCS 3844*, (2006).
24. XML Schema Part 1: Structures Second Edition: http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/
25. XML Schema Part 2: Datatypes Second Edition: http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/

An Abstract Transaction Model for Testing the Web Services Transactions

Rubén Casado[1], Javier Tuya[1], Muhammad Younas[2]

[1]Department of Computing, University of Oviedo, Gijón, Spain
{rcasado, tuya}@uniovi.es
[2]Department of Computing and Communication Technologies, Oxford Brookes University, Oxford, UK
m.younas@brookes.ac.uk

Abstract. This is a summary of the paper published in the Proceedings of the 9th IEEE International Conference on Web Services (ICWS), ranked as CORE A. The work addresses the topic of testing web services transactions using a model-based approach.

Keywords: web services testing, model-based testing

1 Summary

A transaction is defined as a set of operations of an application such that all the operations achieve a mutually agreed outcome. The conventional method for achieving such outcome is the enforcement of the Atomicity, Consistency, Isolation and Durability (ACID) properties. Web Services (WS) transactions are more complex as they involve multiple parties, span many organizations, and may take a long time to finish. Strictly enforcing the ACID properties is not appropriate to a loosely coupled world of WS doing the lock of resources unsuitable. In order to meet the requirements of WS, various extended transaction models have been adapted. So there are a diversity of transaction models and protocols such as Business Transaction Protocol (BTP) or WS BusinessActivity (WS-BA).

Although transactions have been identified as a key issue in WS environments, current research does not focus on a crucial issue of testing them. We proposed a model-based approach to address such gap. Contributions of this work are: (i) an abstract transaction model that serves as a template for modeling current WS transaction standards. (ii) Automatic generation of abstract test scenarios and map them to different WS transactions standards.

1.1 The Abstract Transaction Model

The abstract model has been developed based on main concepts shown in the literature. Its objective is to be easy to understand as well as capable to pattern the

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 145-146, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

actual web service transaction models. We have used the UML statecharts notation since the model is event-driven (messages between participants).

A web service transaction (*wT*) is a set of activities (subtransactions) executed by different web services (participants) that can take a substantial amount of time to complete. We identify four different roles between the participants.

- *Executor*: a participant responsible for executing and terminating a subtransaction.
- *Coordinator*: it coordinates the *wT* and manages failures and compensations. It also collects the results from the participants in order to provide system with a consistent state after the execution of *wT*.
- *Initiator*: it starts the *wT*. First it requests the coordinator for a transaction context. Then it asks to the others participants to participate in the *wT*.
- *Terminator*: it decides when and how the *wT* has to be finished. Thus it participates in the coordination tasks so it can be a subcoordinator.

The modeling of WS transaction standards is achieved following these three algorithms:

- *Role identification and modeling*: it identifies the roles of participants in a target WS transaction standard and models it using the roles defined in the abstract transaction models.
- *State transitioning*: it captures the important states of the target WS transaction standards and maps them to the state transitions of the abstract model.
- *Message syntax*: it transforms the messages of abstract transaction models to the specific protocols of the WS transaction standards.

1.2 Model-based Testing

The main goal of testing is failure detection i.e., the observable differences between the behaviors of implementation and what is expected. We use model-based testing since our abstract model allows us to pattern the transaction behavior. The steps used in the process of definition of test scenarios are described as follows:

- *Test criterion selection:* The transition coverage criterion defines that the set of test scenarios must include tests that cause every transition in the model to be taken.
- *Generating abstract test scenarios*: An abstract test scenario is defined as a sequence of states and transitions of a participant using the abstract model.
- *Generating specific test scenarios*: An abstract test scenario is transformed to a sequence of messages between participants using a specific WS transaction standard.

Our prototype tool automatically obtains the set of test scenarios. It applies transition coverage criterion over the abstract model and obtains a set of independent paths. Each path defines an abstract test scenario. The tool also generates the mapping from the abstract test scenario to a specific test scenario (sequence of message using the syntax of BTP or WS-BA).

A Model of User Preferences for Semantic Services Discovery and Ranking
(Published in ESWC 2010)

José María García, David Ruiz, and Antonio Ruiz-Cortés

University of Seville
Escuela Técnica Superior de Ingeniería Informática
Av. Reina Mercedes s/n, 41012 Sevilla, Spain
josemgarcia@us.es

Abstract. Current proposals on Semantic Web Services discovery and ranking are based on user preferences descriptions that often come with insufficient expressiveness, consequently making more difficult or even preventing the description of complex user desires. There is a lack of a general and comprehensive preference model, so discovery and ranking proposals have to provide ad hoc preference descriptions whose expressiveness depends on the facilities provided by the corresponding technique, resulting in user preferences that are tightly coupled with the underlying formalism being used by each concrete solution. In order to overcome these problems, in this paper an abstract and sufficiently expressive model for defining preferences is presented, so that they may be described in an intuitively and user-friendly manner. The proposed model is based on a well-known query preference model from database systems, which provides highly expressive constructors to describe and compose user preferences semantically. Furthermore, the presented proposal is independent from the concrete discovery and ranking engines selected, and may be used to extend current Semantic Web Service frameworks, such as WSMO, SA-WSDL, or OWL-S. In this paper, the presented model is also validated against a complex discovery and ranking scenario, and a concrete implementation of the model in WSMO is outlined.

Keywords: User Preferences, Ontology Modeling, Semantic Web Services, Service Discovery, Service Ranking.

Summary of the Contribution

In this paper, published in the 7th Extended Semantic Web Conference (ESWC 2010) [1], we presented a highly expressive model aimed at decoupling user preferences definition from underlying formalisms of discovery and ranking engines. These engines typically offer ad hoc ontologies to define user preferences, constraining the expressiveness and making difficult their combination with other discovery and/or ranking approaches. In order to overcome these issues, we proposed an intuitive preference model based on a strict partial order interpretation of preferences.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 147-148, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

Essentially, our preference ontology offers the user a series of constructs that allow to define (1) concrete *atomic* preference terms, which state preferred values for a particular service property, and (2) *composite* preferences, which allow the composition of different preference terms using intuitive criteria.

Concerning atomic preference terms, our ontology provides both qualitative and quantitative facilities, that can be correspondingly applied to non-numerical and numerical service properties. Each atomic preference term refers to a single property, though they can be combined using composite preferences. Therefore, composite preferences allows the definition of complex preferences regarding several service properties.

In order to evaluate the usefulness of our proposal, we validated our model using a complex scenario about logistics management from the SWS Challenge[1]. This scenario consists on seven logistics service offers, described in natural language in terms of different properties, along with a series of service requests (goals) that contain both hard requirements and user preferences. The performed validation proved that our preference model can be used to define complex user preferences.

Additionally, we discussed the extension of WSMO service goals with preference information using our model. This application allowed a seamless integration of preference definitions in WSMO descriptions, refining the service goal meta-model. Therefore, current discovery and ranking approaches could still be applied to extended goals definitions, whereas specialized ranking engines can be easily implemented to account for preferences.

In conclusion, this research work presented a novel approach to define user preferences for Semantic Web Services that offers a highly expressive, intuitive semantic model, which supports and combines both qualitative and quantitative preference terms. Moreover, our proposal is independent of the underlying discovery and ranking formalisms, allowing its extension and application to any Semantic Web Services framework.

Acknowledgments This work has been partially supported by the European Commission (FEDER) and Spanish Government under CICYT projects Web-Factories (TIN2006-00472) and SETI (TIN2009-07366), by the Andalusian Government under projects ISABEL (TIC-2533) and THEOS (TIC-5906), and by the EC FP7 Network of Excellence 215483 S-CUBE.

References

1. García, J.M., Ruiz, D., Ruiz-Cortés, A.: A model of user preferences for semantic services discovery and ranking. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC (2). Lecture Notes in Computer Science, vol. 6089, pp. 1–14. Springer (2010)

[1] The complete scenario description can be found at `http://sws-challenge.org/wiki/index.php/Scenario:_Logistics_Management`

Aplicación de la técnica de pruebas metamórficas a una composición de servicios: Metasearch

Mª del Carmen de Castro Cabrera, Azahara Camacho Magriñán, e Inmaculada Medina Bulo

Universidad de Cádiz, Escuela Superior de Ingeniería
C/ Chile 1, CP 11002, Cádiz, España,
{maricarmen.decastro, inmaculada.medina}@uca.es,
azahara.camachmagri@alum.uca.es

Resumen Debido a que las técnicas de prueba tradicionales no están adaptadas a las características peculiares de los servicios web, se hace necesario el diseño de nuevas técnicas que ayuden en este ámbito. En un trabajo previo se propuso las pruebas metamórficas como una técnica válida para aplicar a composiciones de servicios web en WS-BPEL. En este trabajo se aplica la arquitectura propuesta allí a la composición de servicios Metasearch, que por su complejidad requiere un análisis detallado. Se incluye el estudio y especificación de las relaciones metamórficas para esta composición. Así mismo, se añade una comparativa de otras composiciones estudiadas que muestra resultados prometedores.

Keywords: pruebas metamórficas, servicios web, WS-BPEL

1. Introducción

El lenguaje WS-BPEL 2.0 [7] posibilita la creación de nuevos servicios web (WS) diseñando procesos de negocio más complejos a partir de otros existentes. Por tanto, es preciso implementar buenos métodos de prueba de composiciones que sean correctos. Avances en este aspecto se describen en [8].
La *prueba metamórfica* (MT) [4] es una técnica de prueba de software que permite generar casos de prueba para verificar programas de manera automática. Se basa en el concepto de *relación metamórfica* (MR), que es una propiedad esperada o existente del software que se prueba y que está definida sobre un conjunto de entradas y sus correspondientes salidas.
Recientemente, se ha publicado un interesante trabajo basado en el análisis de los modelos de características [9].
Este trabajo presenta la aplicación de MT a composiciones de servicios web WS-BPEL mediante la arquitectura propuesta en [3] a un caso de estudio, y una comparativa en referencia a otra técnica y de aplicación a otras composiciones con resultados esperanzadores.
El artículo está estructurado de la siguiente manera: En la sección 2 se describen los conceptos básicos de MT. En la sección 3 se describe un caso de estudio, la composición *Metasearch* o Metabúsqueda, junto con las MR y los resultados

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 149-154, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

obtenidos y una comparativa de la técnica aplicada a otras composiciones. Por último, en la sección 4 se comentan las conclusiones y el trabajo futuro.

2. Preliminares

El *problema del oráculo* es uno de los mayores retos al que se enfrentan las técnicas de prueba de software. Un *oráculo* es una entidad capaz de determinar si un programa bajo prueba tiene el comportamiento esperado durante su ejecución. MT se propone como técnica para aliviar el problema del oráculo [5].
Además, MT está relacionado con la noción de MR. En [1] las MR se definen como *propiedades existentes sobre el conjunto de entradas y sus correspondientes resultados para múltiples evaluaciones de una función.*
Cuando la implementación es correcta, se espera que las entradas y sus correspondientes salidas cumplan determinadas propiedades necesarias que son relevantes a los algoritmos subyacentes.
Una MR debería proveer una forma de generar nuevos casos de prueba a partir de otros dados previamente. Para ilustrar esto, consideremos un programa *inv-orden* que, dada una lista L_1 formada por números naturales, por ejemplo $(2, 4, 5)$, invierte el orden de los elementos de la lista. Al aplicar el programa a dicha lista, obtenemos una nueva lista $L_2 = (5, 4, 2)$ con los elementos en orden inverso. Si multiplicamos cada uno de los elementos de la lista inicial L_1 por un escalar, por ejemplo 3, la lista resultante L_1' es $(6, 12, 15)$, que al introducirla como entrada al programa *inv-orden*, resulta la lista $L_2' = (15, 12, 6)$. Podemos observar que L_2 y L_2' están relacionadas y que se debe cumplir que L_2' es igual a multiplicar cada uno de los elementos de L_2 por 3. Formalmente, podemos obtener una expresión para esta relación metamórfica: $\mathrm{MR}_1 : \exists L_1, L_1', L_2, L_2'$ tal que $L_1' = 3 \cdot L_1 \wedge L_2 = \textit{inv-orden}(L_1) \wedge L_2' = \textit{inv-orden}(L_1') \Rightarrow L_2' = 3 \cdot L_2$

En resumen, MT es una técnica que comienza con un conjunto de prueba inicial, que es obtenido con alguna estrategia de selección de casos de prueba, y un conjunto de MR. Una vez que el programa es ejecutado sobre este conjunto de casos de prueba, se detectan y corrigen errores hasta obtener un *conjunto de casos de prueba exitosos* que contiene casos de prueba que, en principio, no fallan. Las MR utilizan este conjunto exitoso para generar nuevos casos de prueba. Estos casos de prueba constituyen el *conjunto de casos de prueba siguientes* y el proceso es iterado hasta que se cumple el criterio de prueba establecido.

3. Aplicación de MT a un caso de estudio y comparativa

En esta sección se va a aplicar la arquitectura descrita en [3] a la composición MetaSearch definida en [2]. Su lógica puede verse en la Figura 1 y consiste en la búsqueda realizada por un cliente en los buscadores de Google y MSN. Cada uno de los buscadores devuelve sus resultados, primero todos los de Google y luego los de MSN. Si no hay resultados de Google, el cliente sólo verá los resultados devueltos por MSN, y viceversa. Se trata de una composición compleja

si la comparamos con otras como LoanApproval estudiada en [3], ya que MetaSearch posee concurrencia en puntos como en la búsqueda de resultados por parte de ambos buscadores, bucles condicionales para la devolución de resultados, variables internas que intervienen en la evolución de las ejecuciones, etc.

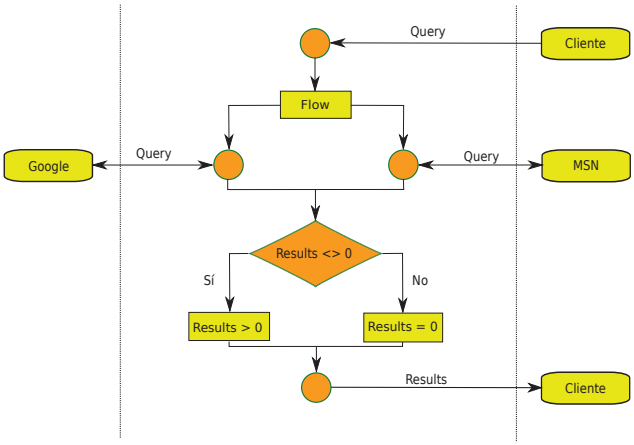


Figura 1: Diagrama de flujo de la composición MetaSearch

La primera fase consiste en analizar la composición y los casos de prueba originales para diseñar MR adecuadas a ellos. Aplicamos una de estas MR a uno de los casos de prueba originales y obtendremos un nuevo caso de prueba que será estudiado posteriormente. Para este primer estudio se utiliza una versión simplificada de la composición en la cual sólo se han considerado los siguientes elementos: *Query*, *Language*, *Country*,*Culture* (su valor es igual a la concatenación de Language y Country; cuando uno de estos elementos es nulo, su valor es la cadena predeterminada 'en-US'), *Max_Results* (número máximo de resultados), *Count* (número total de resultados, $\$Count <= \$Max\_Results$), *Google results*, *MSN results*.
La estructura de un caso de prueba teórico es la siguiente:

(*Query, Language, Country, Max_Results, Count, Google results, MSN results*)

Uno de los casos de prueba originales que hemos estudiado es el siguiente:

(Philipp Lahm, de, DE, 3, 3, *Google results, MSN results*)
$Culture = de\text{-}De$

Aplicamos la siguiente relación metamórfica al caso de prueba previo donde los elementos acabados en '1' se refieren al caso de prueba original y los elementos que acaban en '2' se refieren al nuevo caso de prueba generado:

Precondición: $Language1 \neq "" \wedge Culture2 \neq "en-US"$
MR1: $Language2 = "" \Rightarrow Culture2 \neq Culture1$

La precondición de esta MR indica que el elemento Language1 debe ser no nulo y que tampoco se le haya asignado a Culture2 el valor "en-US". Como podemos ver, el caso de prueba original cumple esta condición, por tanto, si aplicamos la MR1 obtenemos el siguiente caso de prueba:

(Philipp Lahm, , DE, 3, 3, *Google results, MSN results*)
Culture = en-US

Para automatizar la generación de casos de prueba siguientes se ha implementado la MR anterior junto a otras obtenidas del estudio completo. Estas nuevas MR nos permiten obtener nuevos casos de prueba, lo que mejora el conjunto original. La aplicación de MR a ciertos casos de prueba puede generar casos de prueba ya existentes. Para evitar esto, se ha implementado una aplicación que automatiza la generación descartando aquellos que ya existen.

3.1. Evaluación de resultados

Como resultado general de este estudio indicamos que se han conseguido generar nuevos casos de prueba automáticamente. Es decir, se ha mejorado la técnica de detección de errores completando el conjunto de casos de prueba original y detectando nuevos errores que no se habían detectado anteriormente. Consideremos de nuevo el caso de prueba original de la sección anterior y el caso de prueba generado a partir de aplicar MR1.
Basándonos en la lógica de la composición MetaSearch, el fragmento de código que evalúa el resultado final del elemento *Culture* es el siguiente:

```
<bpel:condition>
   (($inputVariable.payload/client:country != '')
      and
   ($inputVariable.payload/client:language != ''))
  </bpel:condition>
```

Como se ha explicado al comienzo de esta sección, si los elementos *Language* y *Country* no son nulos (' ') el valor de *Culture* será la cadena formada por la concatenación entre ambos elementos. En otro caso, el valor que tomará será el predeterminado, 'en-US'.
Supongamos que existe algún error en el código de la composición, como por ejemplo, la sustitución del operador lógico **'and'** por el operador **'or'**. El fragmento de código correspondiente a este error sería el siguiente:

```
<bpel:condition>
   (($inputVariable.payload/client:country != '')
      or
   ($inputVariable.payload/client:language != ''))
  </bpel:condition>
```

Ahora aplicamos el caso de prueba original y el generado a este fragmento erróneo para ver si realmente conseguimos detectar el error. Con el caso de prueba original no detectamos la composición errónea porque también satisface la condición

modificada. Sin embargo, si usamos el nuevo caso de prueba generado con la MR1, detectamos la composición errónea.
En la composición correcta, el valor del elemento *Culture* del nuevo caso de prueba es 'en-US' ya que uno de los elementos (en este caso, el elemento *Language*) es igual a nulo. Sin embargo, en la composición errónea el valor de *Culture* es '-DE' debido a que este caso de prueba satisface la condición de que al menos uno de los elementos sea diferente de nulo. Como tenemos resultados diferentes para el elemento *Culture*, el error es detectado. Por tanto, además de generar un número importante de nuevos casos de prueba automáticamente, muchos de estos casos nos permiten detectar nuevos errores y mejorar el conjunto de casos de prueba original. Por otro lado, podemos implementar nuevas MR que nos permitan generar nuevos casos de prueba que mejoren los resultados.

3.2. Comparativa de aplicación de MT a composiciones WS-BPEL

La generación de nuevos casos de prueba a partir de las MR ofrece unos resultados relevantes al detectar errores que anteriormente no se habían detectado con el conjunto de casos de prueba original. En este apartado se muestran los resultados obtenidos de la composición MetaSearch y otras dos composiciones más a partir de la aplicación de las MR al conjunto de casos de prueba original.

En la Tabla 1 se muestra la comparativa de los resultados iniciales utilizando la técnica de prueba de mutaciones con la herramienta MuBPEL [6] y los resultados posteriores utilizando los casos de prueba generados con las MR.

Tabla 1: Comparativa del número de mutantes muertos y de casos de pruebas generados

Composición	**LoanApproval**	**MarketPlace**	**MetaSearch**
Mutantes generados	93	34	706
Casos de prueba iniciales	14	9	7
Casos de prueba siguientes	188	656	1784
Mutantes muertos previos a MT	82	33	566
Mutantes muertos después de MT	86	34	619
MR generadas	12	13	15

Como podemos ver en las tres composiciones se mejoran los mutantes muertos y, en concreto, en MetaSearch se matan 53 mutantes más. En todas ellas se genera un gran número de casos de prueba respecto al número de MR implementadas (en el caso de Metasearch la diferencia es bastante más significativa debido al tipo de composición y a las MR usadas en ese caso), ya que cuando se cumple la precondición de una MR, se aplicará sobre todos los casos de prueba ya sean originales u obtenidos por la aplicación previa de alguna otra MR. Y la razón por la que se obtiene un número de mutantes muertos no demasiado grande frente al número de casos de prueba generados es debido a que estos son casos de prueba generales y generados automáticamente, y no todos ellos matan mutantes.

4. Conclusiones

Es importante desarrollar técnicas que permitan probar software para sevicios web, en especial para composiciones en WS-BPEL. Por otro lado, MT ha sido implementada en diferentes lenguajes de manera eficiente y es objeto de estudio por parte de diversos grupos en la actualidad. Basándonos en la arquitectura propuesta en un trabajo previo se ha incluido un caso de una composición concreta, *MetaSearch*. Se describen la especificación, diseño y descripción de la implementación, así como los resultados obtenidos. Se muestra la forma de mejorar un conjunto de casos de prueba para que se detecten más errores.

Como trabajo futuro, está el desarrollo completo de la arquitectura propuesta. Una vez esté implementado el sistema, se podría comparar los resultados con los obtenidos con otras técnicas usando distintas composiciones.

Agradecimientos

Este trabajo ha sido financiado por MoDSOA (TIN2011-27242) del Programa Nacional I+D+i del Ministerio de Ciencia e Innovación.

Referencias

1. Andrews, J.H., Briand, L.C., Labiche, Y.: Is mutation an appropriate tool for testing experiments? In: Proceedings of the 27th International Conference on Software Engineering (ICSE 2005). pp. 402–411. ACM Press (2005)
2. BPELUnit: Metasearch. http://bpelunit.sourceforge.net/example.html
3. Castro-Cabrera, M.d.C., Camacho-Magriñán, A., Medina-Bulo, I., Palomo-Duarte, M.: Una arquitectura basada en pruebas metamórficas para composiciones de servicios ws-bpel. In: Actas de las VII Jornadas de Ciencia e Ingeniería de Servicios. pp. 9–22. Servizo de publicacións da Universidade da Coruña, A Coruña, España (Sep 2011)
4. Chen, T.Y.: Metamorphic testing: A new approach for generating next test cases. HKUSTCS98-01 (1998)
5. Chen, T.Y.: Metamorphic testing: A simple approach to alleviate the oracle problem. In: Proceedings of the 5th IEEE International Symposium on Service Oriented System Engineering. IEEE Computer Society (2010)
6. García-Domínguez, A., Estero-Botaro, A., Domínguez-Jimenez, J.J., Medina-Bulo, I.: Mubpel: una herramienta de mutación firme para ws-bpel 2.0. In: aceptado para JISBD 2012. Almería, Spain (2012)
7. OASIS: Web Services Business Process Execution Language 2.0. http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html (2007), Organization for the Advancement of Structured Information Standards
8. Palomo-Duarte, M.: Service composition verification and validation. In: Jonathan Lee, S.P.M., Liu, A. (eds.) Service Life Cycle Tools and Technologies: Methods, Trends and Advances, pp. 200–219. IGI Global (2011)
9. Segura, S., Galindo, J., Benavides, D., Parejo, J., Ruiz-Cortés, A.: Betty: Benchmarking and testing on the automated analysis of feature models. In: Eisenecker, U., Apel, S., Gnesi, S. (eds.) Sixth International Workshop on Variability Modelling of Software-intensive Systems (VaMoS'12). pp. 63–71. ACM, Leipzig, Germany (2012)

Sesion 5

SOA, Tecnologías para Servicios Web y Aplicaciones II

Chair: *Dr. Víctor Ayllón*

Sesion 5: SOA, Tecnologías para Servicios Web y Aplicaciones II
Chair: Dr. Víctor Ayllón

Carlos Müller, Marc Oriol Hilari, Marc Rodríguez, Xavier Franch, Jordi Marco, Manuel Resinas and Antonio Ruiz-Cortés. *SALMonADA: A Platform for Monitoring and Explaining Violations of WS-Agreement-Compliant Documents*.

José María García, David Ruiz and Antonio Ruiz-Cortés. *SOA4All Integrated Ranking: A Preference-based, Holistic Implementation*.

José A. Martin, F. Martinelli and Ernesto Pimentel. *Synthesis of Secure Adaptors*.

Jose A. Dorado, Juan Boubeta-Puig, Guadalupe Ortiz and Inmaculada Medina-Bulo. *Detección de Ataques de Seguridad mediante la Integración de CEP y SOA 2.0*.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): Actas de las "VIII Jornadas de Ciencia e Ingeniería de Servicios (JCIS'2012)", Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

SALMonADA: A Platform for Monitoring and Explaining Violations of WS–Agreement–Compliant Documents*

C. Müller[1], M. Oriol[2], M. Rodríguez[2], X. Franch[2], J. Marco[2], M. Resinas[1], and A. Ruiz–Cortés[1]

[1] University of Seville, LSI
ISA research group, http://www.isa.us.es/, Seville (Spain)
{cmuller,resinas,aruiz}@us.es
[2] Technical University of Catalunya
GESSI research group, http://www.essi.upc.edu/~gessi/, Barcelona (Spain)
{moriol,jmarco}@lsi.upc.edu, {mrodrigues,franch}@essi.upc.edu

Abstract. Several research frameworks in both academy and industry aim at monitoring conditions stated in Service Level Agreements (SLAs). However, up to our knowledge, none of them present reasoning capabilities over the SLA with a clear explanation of the concrete statements that violate the agreement. In this paper we present SALMonADA, a platform to monitor SLAs specified with WS–Agreement , that provides agreement violations explanations by pointing both: violated terms of the WS–Agreement document, and violating measures of a monitoring management document.

1 Problem Motivation

There is a real need to have infrastructures and Service Based Systems (SBS) regulated by Service Level Agreements (SLAs). WS–Agreement is arguably the most widespread recommendation for defining SLAs. However, most developing environments do not include enough matured facilities to develop SLA–driven applications in general or WS–Agreement –driven applications in particular. The urgency to overcome this situation may change overnight if we consider the need for SLAs to regulate the increasing number of things that can be delivered as services over the Internet (the well–known XaaS or everything as a service). As an example, we find that several research frameworks in both academy and industry aim at monitoring conditions stated in SLAs to detect violations. However, up to our knowledge, between them there are only a couple of works providing monitoring facilities for WS–Agreement documents. Moreover, none of the proposals present reasoning capabilities over the SLA with a clear explanation of the concrete statements that violate the agreement. This lack of proposals to explain

* This work has been partially supported by: S–Cube, the European Network of Excellence in Software Services and Systems; the European Commission (FEDER); the Spanish Government under the CICYT projects SETI (TIN2009–07366) and ProS–Req (TIN2010–19130–C02–01); and by the Andalusian Government under the projects THEOS (TIC–5906) and ISABEL (P07–TIC–2533).

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 157-160, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

violations in WS–Agreement documents pushed us to propose SALMonADA, a SBS comprising the monitoring capabilities of SALMon-aaS[3] and the analysis capabilities of ADA-aaS[4] , with the following architecture, novelties and functions.

2 Novelties

The novelty of our proposal is given by the following SALMonADA features:

Full WS–Agreement compliant. That is, it supports WS–Agreement documents with expressive terms including arithmetic-logic expressions relating several metrics inside the service level objectives (SLOs), and some elements not supported by other monitoring proposals yet, as far as we know, such as: (1) term compositors defining agreement variants inside an agreement; (2) several terms scopes denoting the affected service operations; and (3) qualifying conditions inside terms to enable or disable the SLO.

SLOs violations explanation-aware considering the afore mentioned expressive SLOs. That is, it provides both for SLOs violations: violated terms of the WS–Agreement document, and violating measures got at monitoring time and stored in a monitoring management document. For instance, an average response time may be calculated for each service operation allowing to include a SLO for each operation as follows: "AverageResponseTime <= x sec". When a SLO violation is detected at monitoring, only such term which scopes to the violated service operation and which includes the violated SLO would be returned as violation explanation[5].

Asynchronous/Synchronous-compliant. SALMonADA platform is designed and developed to support asynchronous and synchronous interaction styles with their clients. Thus, a client, based on its own benefit, may choose its preferred approach. Independently of the selected approach a client must start and stop the SALMonADA monitoring to be subscribed/unsubscribed as client.

Decoupled Monitoring Management. That is, it extracts a monitoring management document (MMD) as a monitoring view from a WS–Agreement document. Thus, SALMonADA comprised services use WS–Agreement documents and MMDs as operation inputs/outputs.

Extensible. to monitor new metrics by decoupling monitoring logic for each quality metric, making it easily upgradeable. Currently SALMonADA is extensible at deployment time but it will be so at run time in a nearby future.

3 SALMonADA Architecture

As shown in Fig. 1, we have developed SALMonADA as a SBS with the following elements:

[3] `gessi.lsi.upc.edu/salmon/web/`
[4] `www.isa.us.es/ada/`
[5] This scenario is included in a violating SLA test case of the web application developed as SALMonADA client

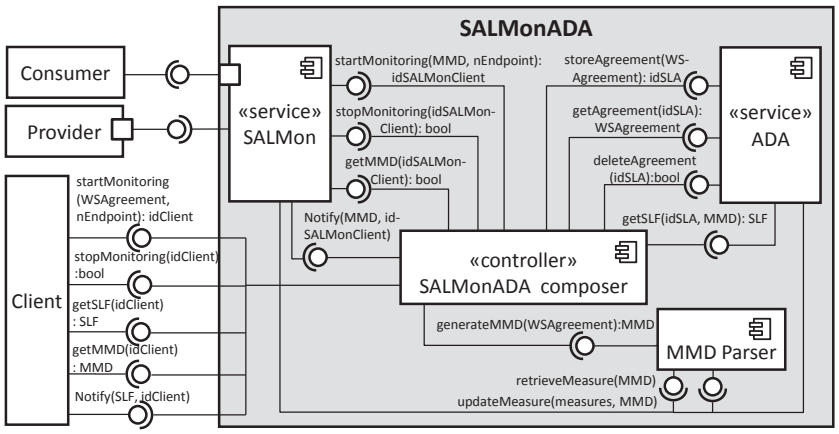


Fig. 1. Architectonic Model of SALMonADA

Client provides the SLA to monitor expressed in WS–Agreement. It is able to retrieve either the MMD or the analysis result, and if desired, it can also receive asynchronously notifications when the SLA has been violated.

[carlos:] Client a la izquierda

SALMonADA composer is the service that composes the internal services of the platform. It provides the interface to the client and manages the execution process of the system. It also adds an independence layer on the interaction required between the analysis of the SLAs (performed by ADA-aaS) and the monitoring of the services QoS (performed by SALMon-aaS). Such a decoupled structure allows to add or modify the internal components in a very flexible manner. (i.e. allows to replace the monitor or the analyzer without affecting the other elements of the platform).

SALMon is the service responsible for monitoring the services QoS.

ADA is the service responsible of managing and analyzing the WS–Agreement documents. It supports the analysis of WS–Agreements with expressive assertions inside guarantee terms.

MMD Parser is the service that extracts the MMD from the SLA, and it also implements the functionality to interact with the MMDs (retrieve or update values). Thus, the MMD structure, whose information is used by all platform components, is decoupled from both ADA and SALMon. Therefore, different MMD structures can be developed, if needed.

4 Functions

We have developed a web application for SALMonADA[6] for demonstration purposes that supports afore mentioned novelties. For instance, Fig. 2 depicts the web appli-

[6] it can be tried at `www.isa.us.es/ada.source/SLAnalyzer/` and a screencast is available at `gessi.lsi.upc.edu/salmon/ada/`

cation highlighting as violation explanation that the `AverageResponseTime` of `explainNonCompliance` operation is the violating metric because it was measured as `3.421` seconds, while the ADA-aaS SLA guarantee term obligates the provider to respond in less than `2` seconds.

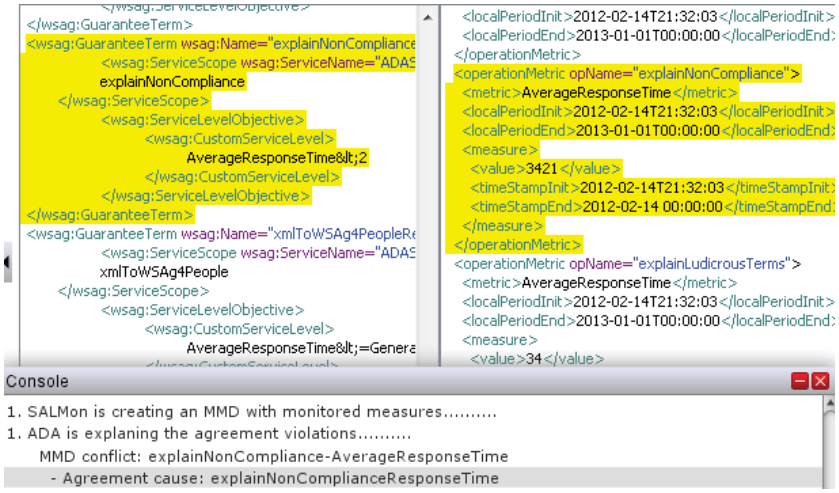


Fig. 2. Reporting a violation with SALMonADA

The web application provides also some additional functions such as: `SLA violation checking` that informs about the agreement violating state without providing any kind of explanation; `violating and non-violating WS-Agreement documents` of SALMon-aaS and ADA-aaS, to test the platform; and an `edition panel` to: (1) change/edit the SLA to be monitored and analysed; (2) show the MMD without and with monitored measures; and (3) show in a parallel view the SLOs violation explanation including both: the SLA highlighting violated terms, and the MMD highlighting the monitored violating measures. Finally, the web application also includes a `log console` to show the overall process.

SOA4All Integrated Ranking
A Preference-based, Holistic Implementation

José María García, David Ruiz, and Antonio Ruiz-Cortés

University of Seville
Escuela Técnica Superior de Ingeniería Informática
Av. Reina Mercedes s/n, 41012 Sevilla, Spain
josemgarcia@us.es

Abstract. There exist many available service ranking implementations, each one providing ad hoc preference models that offer different levels of expressiveness. Consequently, applying a single implementation to a particular scenario constrains the user to define preferences based on the underlying formalisms. Furthermore, preferences from different ranking implementation's model cannot be combined in general, due to interoperability issues. In this article we present an integrated ranking implementation that enables the combination of three different ranking implementations developed within the EU FP7 SOA4All project. Our solution has been developed using PURI, a Preference-based Universal Ranking Integration framework that is based on a common, holistic preference model that allows to exploit synergies from the integrated ranking implementations, offering a single user interface to define preferences that acts as a façade to the integrated ranking implementation.

Keywords: Semantic Web Services, Ranking Tools, Systems Integration, Preference Models

1 Introduction

Within the EU FP7 SOA4All project[1], three different ranking implementations were implemented [6], offering users different choices depending on their expressiveness and performance requirements for the service ranking process. Firstly, a simple, yet efficient objective ranking mechanism provides some metrics about the quality of service and its description. Secondly, a multi-criteria non-functional property (NFP) based ranking allows a more expressive definition of preferences on non-functional properties. Finally, a fuzzy logic based ranking implementation offers a highly expressive solution to define preferences, though the ranking process is less performant.

In order to take full advantage of the three developed ranking techniques, a user should be able to express preferences using every facility those ranking techniques provide, at the same time. Therefore, at the final stage of SOA4All project, an integrated ranking approach has been developed, so that a user can define and compose preferences using a generic and expressive model that integrate preference definitions used in the other ranking techniques. This integrated ranking approach can be viewed as a holistic façade to access available ranking techniques using a common, unique access point to them. SOA4All Integrated Ranking is available online at http://www.isa.us.es/soa4all-integrated-ranking/

[1] http://www.soa4all.eu

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 161-164, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

2 Preference Modeling

The preference model used in this approach is an adaptation of a comprehensive, user-friendly model described in [3]. Basically, the user can express atomic preferences using different preference terms that are handled internally by the corresponding ranking approach, and then composite preferences can be used to compose those terms, defining the relationship between previously expressed atomic preferences. Figure 1 shows a UML representation of this preference model.

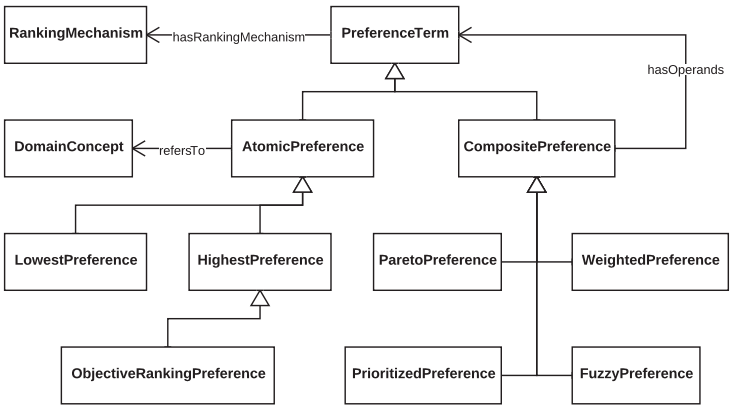


Fig. 1. Simplified UML representation of the preference model.

Essentially, each preference term is handled by a corresponding ranking mechanism, namely objective ranking metrics, multi-criteria NFP-based ranking, and fuzzy logic based ranking, while more generic composite preferences are directly handled by the integrated ranking framework used in the implementation (see Sec. 3). Note that fuzzy preferences representation is simplified in the diagram (see [2] for a more detailed description). The correspondences between preference terms and ranking mechanisms are summarized in Table 1.

Atomic preferences are related to a domain-specific concept that represents a NFP that should be optimized to fulfill the user preference over it. For instance, a `Lowest` (a `Highest`) preference means that the user prefers an NFP value the lower (the higher) the better. These preferences mimic the ascending or descending order defined in the multi-criteria, NFP-based ranking approach, while using `Weighted` preferences the user can define each atomic preference interest value.

The objective ranking metrics approach is actually an optimization of ranking metrics, so it is handled similarly to a highest preference, but the referred domain concept to optimize is one of the available metrics. Finally, users can compose preferences by balancing their fulfillment degree (a Pareto preference) or prioritizing some preferences over others (a Prioritized preference). See [3] for further details.

Table 1. Correspondences between ranking mechanisms and preference terms.

Preference Term	Ranking Mechanism
`LowestPreference`	MultiCriteriaRanking
`HighestPreference`	MultiCriteriaRanking
`ObjectiveRankingPreference`	ObjectiveMetricsRanking
`ParetoPreference`	DefaultParetoRanking
`PrioritizedPreference`	DefaultPrioritizedRanking
`WeightedPreference`	MultiCriteriaRanking
`FuzzyPreference`	FuzzyLogicBasedRanking

3 SOA4All Integrated Ranking Implementation

The developed integrated preference-based ranking approach evaluates preferences defined after the presented model in order to rank a set of discovered services. As described before, each preference term is handled by a particular ranking mechanism. In order to correctly call each mechanism, compose the results, and manage in general the integrated ranking process, the implementation is based on the PURI[2] framework [2]. PURI provides facilities to integrate several ranking mechanisms by using an extended preference model, that can be also used to streamline the previous discovery process [4]. The implemented ranking solution adapts the PURI framework, integrating the three ranking approaches developed in SOA4All [1].

This implementation is published as a web service that provides a method that receives a set of services to rank and the user preference defined after the discussed preference model. Concretely, this method firstly analyses the user preference term. Then, service ranking for each preference term is delegated to the corresponding ranking mechanism presented in Table 1. The adaptation of PURI framework that has been developed is responsible to both the delegation mechanism and the composition of ranked results for each preference term. Finally, the method returns the requested ranked list of services.

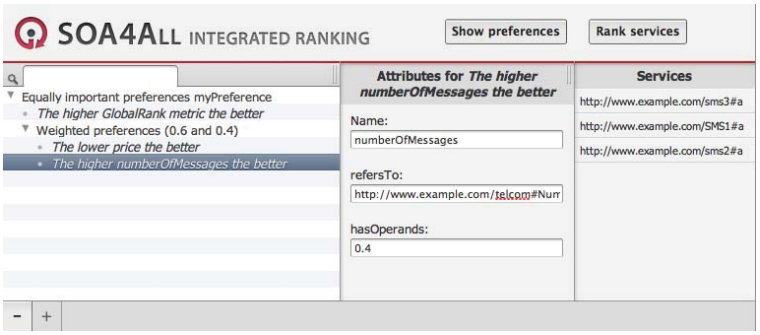


Fig. 2. Screenshot of the preference definition user interface.

[2] An early prototype described in [5] can be found at `http://www.isa.us.es/upsranker`

Furthermore, a user interface to define preferences and rank services accordingly have been developed, using the Google Web Toolkit and based on AcME modeling toolkit[3]. This interface allows the user to easily define preferences based on the discussed model. For instance, in Figure 2, a user has defined a preference that balance the importance of a higher global rank with a multi-criteria preference over a lower price (with an interest value of 0.6) and a higher number of delivered messages (with an interest value of 0.4). Additionally, the interface can also be used to test the integrated preference based ranking implementation, so a set of pre-loaded services can be ranked in terms of the created preferences, using the "Rank services" button.

4 Conclusions

Our presented tool implementation, SOA4All Integrated Ranking, offers a holistic solution to integrate several ranking implementations that provides users with the flexibility to choose and combine any of the preference facilities offered by the other three ranking mechanisms proposed within SOA4All project, making the most of them by exploiting their synergies. Nevertheless, a single user interface for accessing the whole ranking process simplifies the user interaction with the SOA4All discovery and ranking solution. Finally, additional ranking mechanisms may be also integrated with our solution, by identifying corresponding preferences from our common model and implementing an adapter that would be automatically instantiated by PURI.

Acknowledgments This work has been partially supported by the European Commission (FEDER) and Spanish Government under CICYT project SETI (TIN2009-07366), by the Andalusian Government under projects ISABEL (TIC-2533) and THEOS (TIC-5906), by the EU FP7 IST project 27867 SOA4All, and by the EC FP7 Network of Excellence 215483 S-CUBE.

References

1. Agarwal, S., Junghans, M., Norton, B., García, J.M.: Second service ranking prototype. Deliverable 5.4.3, SOA4All (2011)
2. García, J.M., Junghans, M., Ruiz, D., Agarwal, S., Ruiz-Cortés, A.: Integrating semantic web services ranking mechanisms using a common preference model. Knowledge-Based Systems (2012), in press.
3. García, J.M., Ruiz, D., Ruiz-Cortés, A.: A model of user preferences for semantic services discovery and ranking. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC (2). Lecture Notes in Computer Science, vol. 6089, pp. 1–14. Springer (2010)
4. García, J.M., Ruiz, D., Ruiz-Cortés, A.: Improving semantic web services discovery using sparql-based repository filtering. Web Semantics: Science, Services and Agents on the World Wide Web (2012), in press.
5. García, J.M., Toma, I., Ruiz, D., Ruiz-Cortés, A.: A service ranker based on logic rules evaluation and constraint programming. In: de Paoli, F., Toma, I., Maurino, A., Tilly, M., Dobson, G. (eds.) NFPSLA-SOC'08. CEUR Workshop Proceedings, vol. 411 (2008)
6. Toma, I., Steinmetz, S., Lausen, H., Agarwal, S., Junghans, M.: First Service Ranking Prototype. Deliverable 5.4.1, SOA4All (2011)

[3] `http://www.isa.us.es/acme/`

Synthesis of Secure Adaptors ⋆,⋆⋆

J.A. Martín[1], F. Martinelli[2], and E. Pimentel[1]

[1] E.T.S. Ingeniería Informática, Universidad de Málaga, Málaga, Spain
{jamartin,ernesto}@lcc.uma.es
[2] Istituto di Informatica e Telematica, National Research Council, Pisa, Italy
Fabio.Martinelli@iit.cnr.it

Security is considered one of the main challenges for software oriented architectures (SOA) [1,2]. For this reason, several standards have been developed around WS-Security. However, these security standards usually hinder interoperability, one of the main pillars of Web service technologies. *Software adaptation* [3] is a sound solution where an adaptor is deployed in the middle of the communication to overcome signature, behavioural and QoS incompatibilities between services. This is particularly important when dealing with stateful services (such as Windows Workflows or WS-BPEL processes) where any mismatch in the sequence of messages might lead the orchestration to a deadlock situation. We proposed *security adaptation contracts* [4] as concise and versatile specifications of how such incompatibilities must be solved. Nonetheless, synthesising an adaptor compliant with a given contract is not an easy task where concurrency issues must be kept in mind and security attacks must be analysed and prevented. In this paper, we present an adaptor synthesis, verification and refinement process based on security adaptation contracts which succeeds in overcoming incompatibilities among services and prevents secrecy attacks. We extended the ITACA toolbox [5] for synthesis and deadlock analysis and we integrated it with a variant of CCS [6], called Crypto-CCS [7], to verify and refine adaptors based on partial model checking and logical satisfiability techniques.

Many standards have been defined to include security in Web services (WSs): XML Encryption, for performing cryptographic operations over parts of XML documents; WS-SecureConversation, to establish secure sessions among services; WS-Trust, to assert and handle trust between different parties; WS-SecurityPolicy, to express the policies that are offered and required; and WS-Security, to glue it all together, to name a few members of the WS-* specifications.

However, the inclusion of these new specifications complicate the development, reuse and replaceability of SOA systems. Security adaptation contracts (SACs) were proposed to solve this problem. SACs are able to abstract, in a concise manner, the information scattered among several WS-* specifications so that it is easy to express a match between the operations offered and required by the services. Additionally, SACs are enhanced with powerful synthesis, verification and refinement algorithms that support the automatic generation of secure orchestrators among incompatible services.

⋆ This is an abstract of an article published in the Journal of Logic and Algebraic Programming (JLAP), volume 81, issue 2, pp. 99-126, Elsevier, February 2012 doi:10.1016/j.jlap.2011.08.001.

⋆⋆ Work partially supported by EU-funded project FP7-231167 CONNECT and by EU-funded project FP7-256980 NESSOS, project P06-TIC-02250 funded by the Andalusian local government and project TIN2008-05932 funded by the Spanish Ministry of Education and Science (MEC) and FEDER.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 165-166, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

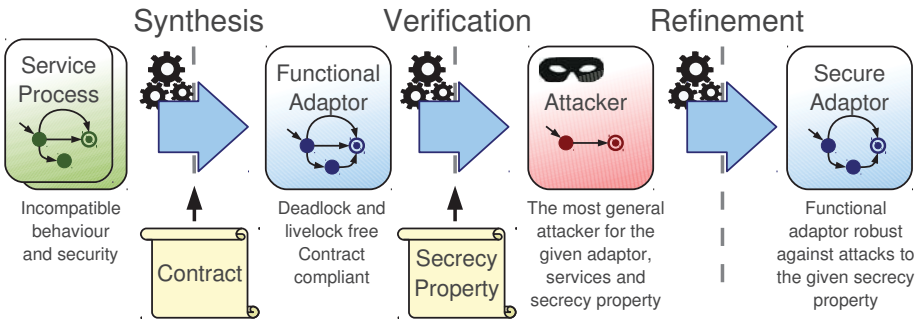


Fig. 1. Overview of the synthesis of secure adaptors

Figure 1 shows an overview of the approach presented in this paper. The inputs of our synthesis process are: i) services with incompatible behaviour and security QoS encoded in Crypto-CCS (CCS extended with cryptographic operations); ii) a security adaptation contract, i.e., a mapping between the interfaces of the services which specifies how the incompatibilities must be solved and which security checks must be enforced; and iii) a secrecy property to preserve expressed in a logical language with knowledge operators. The synthesis process is structured in three sequential steps. First, a *functionally-correct adaptor* is synthesised based on the given contract and services. This adaptor is able to orchestrate the services in a way that it solves their initial incompatibilities and avoids deadlocks and livelocks. However, the adapted system might be insecure against secrecy attacks. For this reason, we verify if the synthesised adaptor and the given services are robust with regard to the given secrecy property in a second step. Finally, if an attack exists, we proceed to refine the initial adaptor into a *secure adaptor*. The final output of the synthesis process presented in this paper is an adaptor encoded in Crypto-CCS, able to orchestrate the services despite their incompatibilities, compliant with the security adaptation contract (which allows a fine-grained control over the result) and robust against attacks to the given secrecy property.

References

1. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-Oriented Computing: State of the Art and Research Challenges. Computer **40**(11) (2007) 38–45
2. Baresi, L., Nitto, E.D., Ghezzi, C.: Toward open-world software: Issues and challenges. Computer **39**(10) (2006) 36–43
3. Yellin, D.M., Strom, R.E.: Protocol Specifications and Components Adaptors. ACM Transactions on Programming Languages and Systems **19**(2) (1997) 292–333
4. Martín, J.A., Pimentel, E.: Contracts for Security Adaptation. JLAP **80**(3-5) (2011) 154 – 179
5. Cámara, J., Martín, J.A., Salaün, G., Cubo, J., Ouederni, M., Canal, C., Pimentel, E.: ITACA: An Integrated Toolbox for the Automatic Composition and Adaptation of Web Services. In: Proc. of ICSE'09, IEEE Computer (2009) 627–630
6. Milner, R.: Communication and concurrency. Prentice-Hall (1989)
7. Martinelli, F.: Analysis of security protocols as open systems. TCS **290**(1) (2003) 1057–1106

Detección de Ataques de Seguridad mediante la Integración de CEP y SOA 2.0

Jose Antonio Dorado Cerón, Juan Boubeta Puig, Guadalupe Ortiz e Inmaculada Medina Bulo

Departamento de Ingeniería Informática, Universidad de Cádiz,
C/Chile 1, 11002 Cádiz, España
jose.doradoce@alum.uca.es
{juan.boubeta,guadalupe.ortiz,inmaculada.medina}@uca.es

Resumen La seguridad informática cada día cobra mayor importancia debido al incremento de ataques que se realizan tanto para intentar acceder a los datos críticos como para detener procesos esenciales en los sistemas. Así pues, la detección temprana de estos ataques es fundamental para asegurar la integridad, disponibilidad y confidencialidad de la información. En este artículo desarrollamos un sistema que integra SOA 2.0 junto con un motor de procesamiento de eventos complejos (CEP) y un sistema de detección de intrusiones (IDS) para detectar inmediatamente las amenazas de seguridad que se produzcan en sistemas complejos y heterogéneos, así como ponerlas en conocimiento a los responsables de seguridad. Estos tomarán las medidas oportunas para reducir el impacto de estas situaciones. Los resultados experimentales obtenidos demuestran que nuestro enfoque, que integra SOA 2.0 con CEP e IDS, es una buena alternativa para el campo de la seguridad informática.

Keywords: CEP, seguridad, amenaza, IDS, Snort, SOA 2.0.

1. Introducción

Actualmente, el campo de la seguridad informática cobra cada día mayor importancia, y eso es debido a que cada vez son más los sistemas de información que almacenan datos críticos para sus usuarios. Esto tiene como consecuencia un significativo incremento del número de atacantes. Por ello es necesario buscar una nueva solución capaz de hacer frente a esta problemática.

En el campo de la seguridad es imprescindible minimizar el tiempo de respuesta a los posibles ataques, debido a que una respuesta fuera de los plazos permisivos puede suponer, en la mayoría de los casos, el éxito del atacante.

Así pues, la tecnología que se ha decidido utilizar atendiendo a los requisitos mencionados es el procesamiento de eventos complejos o *Complex Event Processing* (CEP) [5,8]. Gracias a CEP vamos a poder procesar y analizar en tiempo real una gran cantidad de eventos, además de correlacionarlos entre sí y así poder responder a las situaciones críticas producidas en los sistemas de información. El software que se utilizará es un motor CEP denominado Esper [2,9] que provee

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 167-172, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

un lenguaje de procesamiento de eventos o *Event Processing Language* (EPL) para definir los patrones de eventos que detectarán las situaciones críticas.

A la hora de escoger el enfoque en el que nos vamos a basar, tenemos que tener muy en cuenta que, como afirma Boubeta et al. [6], las arquitecturas orientadas a servicios o *Service-Oriented Architecture* (SOA) [14] no son adecuadas, por sí mismas, para el tratamiento de grandes cantidades de eventos en tiempo real. Por tanto, vamos a hacer uso de la arquitectura alternativa propuesta en ese mismo artículo: una integración de las arquitecturas dirigidas por eventos o *Event-Driven Architecture* (EDA) [12] y SOA, combinada con el uso de CEP. Además, vamos a utilizar un bus de servicios empresariales o *Enterprise Service Bus* (ESB) [13] que nos va a permitir llevar a cabo la integración de las diferentes arquitecturas y tecnologías, además de ofrecer otras ventajas como el desacoplamiento o la creación de un sistema mucho más mantenible y escalable. Concretamente vamos a hacer uso de Mule ESB [3,7] que nos proporciona los componentes necesarios para integrar las diferentes tecnologías.

Otro de los puntos de interés de este trabajo es el uso de una herramienta para la monitorización del tráfico de red, concretamente Snort [4], que es un sistema de detección de intrusos o *Intrusion Detection System* (IDS) basado en reglas que pueden ser desarrolladas a medida. En este artículo exponemos cómo hemos llevado a cabo la integración de este IDS (funciona como productor de eventos) con nuestra aplicación y los resultados que se han obtenido detectando intrusiones en tiempo real.

El resto del artículo se estructura de la siguiente manera. En la sección 2 se describe e implementa el caso de estudio en el que utilizamos nuestro sistema para la detección de amenazas. Posteriormente en la sección 3 se enumera una serie de trabajos relacionados y, por último, en la sección 4 se exponen las conclusiones y el trabajo futuro.

2. Caso de Estudio

En los últimos tiempos el número de ataques contra los sistemas de información y las pérdidas originadas por ello se han visto incrementadas de manera significativa [1]. Por tanto, hemos desarrollado un caso de estudio para paliar cuanto antes esta situación detectando estos ataques en tiempo real.

Se ha adaptado al campo de la seguridad informática la arquitectura SOA 2.0 propuesta en [6] con el fin de detectar los ataques más comunes utilizando Snort como productor de eventos. Como consumidor de la información se ha decidido implementar una solución que envía las alarmas de cada ataque detectado mediante un correo electrónico al responsable de la seguridad del sistema.

En cuanto a los ataques posibles a detectar, hemos escogido algunos de los ataques más comunes que intentan explotar algunos puertos específicos. Se han tenido en cuenta los ataques de denegación de servicios (DOS), ataques que utilizan técnicas de ocultación de identidad (e.g., *spoofing*) como el ataque *smurf* o *land*, ataques sobre puertos específicos tales como ataque *supernuke*, ataque al puerto FTP, ataque *flood* al email o el escaneo de puertos TCP.

2.1. Patrones de Eventos Complejos Aplicados a la Seguridad

A continuación vamos a describir en la Tabla 1 los patrones de eventos complejos que permiten detectar los ataques especificados anteriormente. En esta sub-sección se especifican tan solo el nombre de cada patrón, y los parámetros que tenemos en cuenta en cada caso para detectarlos.

Cuando el requisito para detectar un ataque es que en todas las alertas un parámetro sea siempre el mismo se indicará en la tabla con el valor "igual", mientras que si dicho parámetro debe variar en cada alerta lo indicamos especificando "varía" para dicho parámetro.

Ataque	IP origen	IP destino	Puerto origen	Puerto destino	Eventos/min
DOS	Igual	Igual		Igual	10
Smurf	IP destino	Igual		Distinto de origen	2
Land	IP destino	Igual	Puerto destino	Igual	2
Supernuke		Igual		137, 138 o 139	10
Escaneo puertos	Igual	Igual	Igual	Varía	10
Flood a email		Igual		25, 110, 143, 993	10
FTP		Igual		21	10

Tabla 1. Detalle de los parámetros tenidos en cuenta para cada patrón de evento.

2.2. Arquitectura Propuesta

El funcionamiento básico de nuestra arquitectura (véase Figura 1) es el siguiente: el productor de eventos, Snort, generará una alerta cada vez que detecte una situación anómala. A continuación, el sistema capturará en tiempo real cada una de esas alertas cuya información será transformada en eventos que serán enviados al motor de Esper. Finalmente, este motor correlacionará y comparará los eventos recibidos con los patrones de eventos y, cuando se cumplan las condiciones establecidas en dichos patrones, se informará al *listener* correspondiente que se encargará de notificar estas situaciones al consumidor de eventos pertinente.

2.3. Resultados

En este trabajo se han llevado a cabo los experimentos necesarios para demostrar que el sistema diseñado utilizando Snort y el motor CEP de Esper es adecuado para la detección de ataques de seguridad en tiempo real. Para ello, hemos comprobado qué tasa de eventos es capaz de responder correctamente.

Por lo tanto, para medir la eficiencia del sistema, nuestro objetivo se centra en comprobar cuántas alertas de cualquier ataque es capaz de procesar nuestro sistema. En este caso, estos eventos pueden referirse a cualquiera de los ataques especificados.

La finalidad de nuestro software es detectar un ataque utilizando un número bajo de eventos, ya que lo que se pretende es la detección temprana en el tiempo

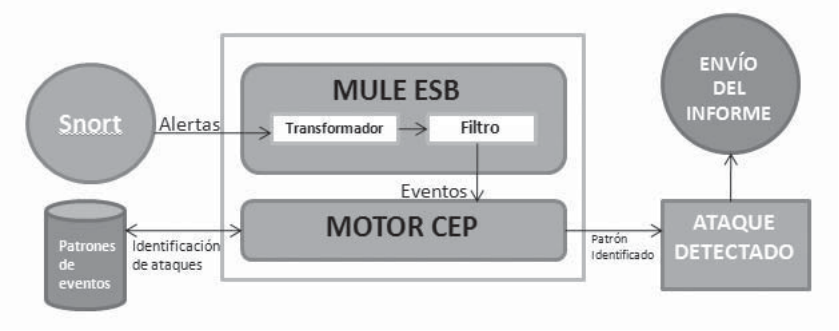


Figura 1. Arquitectura SOA 2.0 que integra Snort, Mule y Esper para la detección temprana de amenazas de seguridad .

de un posible ataque, antes de causar un daño mayor. En un escenario real, a pesar de detectar el ataque con una cifra baja de eventos, los atacantes pueden persistir en el ataque mientras que no se tomen las medidas adecuadas. En la Tabla 2 se muestra el tiempo necesario para procesar los eventos hasta que se detecta la amenaza, así podemos conocer cuál es la capacidad real de nuestro sistema para recibir y procesar grandes cantidades de eventos.

Para obtener un buen rendimiento, este software debe ser ejecutado en una máquina con al menos 2048 MB de memoria RAM y un procesador de 2 GHz. Además, puede ser utilizado tanto en plataformas Windows como Linux.

Para llevar a cabo los experimentos se han generado diferentes baterías de eventos a partir de alertas reales de Snort y se han comprobado cuántos de estos eventos es capaz de procesar y detectar nuestro sistema. Dichas baterías se componen, en cada caso, de una cantidad cada vez mayor de eventos reales de Snort que se pretende que reciba el sistema para comprobar la capacidad de reacción que tiene nuestro sistema en cada uno de estos escenarios.

Nº de eventos recibidos	10	100	500	1000	2000	5000
Tiempo (s)	0,468	0,771	1,895	3,194	11,538	84,184

Tabla 2. Tiempo desde que recibe el primer evento hasta que informa al consumidor.

3. Trabajos Relacionados

Cada vez son más los trabajos que se llevan a cabo sobre CEP. En los últimos tiempos debemos destacar el trabajo de Zappia et al. [15], que proponen una arquitectura muy ligera, fácil de usar, escalable, extensible y portable basada en un diseño de arquitecturas en capas. Estas características son similares a las de nuestra aplicación, gracias sobre todo al uso de un ESB. Este artículo presenta un caso sobre el seguimiento de mercancías peligrosas en el transporte

marítimo y los resultados obtenidos para este campo crítico apoyan nuestro trabajo demostrando que este enfoque es correcto para estas aplicaciones.

En el campo de la seguridad informática podemos destacar el trabajo de Kou y Wen [11] que proponen un sistema de defensa para un *smartphone* utilizando Snort. Este artículo demuestra la portabilidad de este tipo de sistemas, que pueden diseñarse para otro tipo de plataformas como Android. A diferencia de nuestro trabajo, los autores de este artículo no incluyen resultados concluyentes respecto a la capacidad de respuesta del sistema ante grandes cantidades de amenazas, algo que nosotros hemos solucionado con el uso de un motor CEP.

Otro estudio a tener en cuenta es el de Ismail et al. [10] que proponen la implementación de un IDS basado en Snort con el fin de detectar el *malware* en una red de área local. Este estudio se ha implementado en la universidad de Kuala Lumpur y tiene como objetivo la detección de accesos a una base de datos MySQL, WINPCAP y *scripts* de Perl. Aunque a diferencia de nuestra propuesta, no trata de detectar ataques en los que se envíen una gran cantidad de información contra un puerto determinado ni muestra resultados sobre experimentos donde se reciban cantidades importantes de ataques al mismo tiempo.

Existen otros trabajos que proponen el uso de Sn*ort* junto a otras herramientas; por ejemplo, Jiqiang y Yining [16] han diseñado un sistema en el que cooperan tanto Snort como *IPSec* para la detección de diferentes amenazas en sistemas de tipo Windows, a diferencia de nuestro software que sí puede utilizarse en diferentes plataformas.

4. Conclusiones y Trabajo Futuro

En este artículo se ha propuesto el uso de una arquitectura software SOA 2.0 junto con el motor CEP de Esper, utilizando el ESB Mule para la integración de las diferentes tecnologías. Además, hemos utilizado Snort como proveedor de la información y un servidor de correo electrónico como consumidor de la información. Gracias a este enfoque podemos detectar en un reducido espacio temporal distintos ataques de seguridad, que hemos definido previamente mediante patrones de eventos en EPL.

Los resultados demuestran que el uso de CEP es adecuado para trabajar en el campo de la seguridad informática, ya que nos permite trabajar en tiempo real con una gran cantidad de eventos sin ver afectado su rendimiento.

Una de las principales mejoras a introducir en nuestro sistema es la incorporación de nuevos patrones de eventos, que permitirán detectar más ataques de seguridad de los que tenemos en cuenta actualmente.

Aprovechando la escalabilidad del sistema que hemos diseñado, sustituiremos el IDS Snort por otras soluciones disponibles en el mercado. Así podremos observar el comportamiento del sistema ante un cambio de proveedor de información, además de llevar a cabo una comparativa entre los resultados obtenidos actualmente con Snort y los que se puedan obtener utilizando otros productores de eventos. A su vez, también se estudia la posibilidad de combinar varios ESB

y motores CEP para obtener un sistema de procesamiento de eventos a gran escala y más eficiente.

Agradecimientos. Este trabajo fue financiado por el proyecto MoDSOA (TIN 2011-27242) del Programa Nacional de Investigación, Desarrollo e Innovación del Ministerio de Ciencia e Innovación y por el proyecto PR2011-004 del Plan de Promoción de la Investigación de la Universidad de Cádiz.

Referencias

1. Abc (septiembre 2011), http://www.abc.es/20110915/sociedad/abci-perdidas-cibercrimen-201109151628.html
2. Esper (diciembre 2011), http://esper.codehaus.org/
3. Mule ESB (enero 2012), http://www.mulesoft.org/
4. Snort (enero 2012), http://www.snort.org/
5. Ayllón, V., Reina, J.M.: CEP/ESP: Procesamiento y Correlacion de gran Cantidad de Eventos en Arquitecturas SOA. In: 4th Jornadas Científico-Técnicas en Servicios Web y SOA. pp. 97–110. Sevilla (2008)
6. Boubeta Puig, J., Ortiz, G., Medina Bulo, I.: Procesamiento de Eventos Complejos en Entornos SOA: Caso de Estudio para la Deteccion Temprana de Epidemias. In: Actas de las VII Jornadas de Ciencia e Ingenieria de Servicios. pp. 63–76. Servizo de publicacions da Universidade da Coruña, A Coruña, Spain (septiembre 2011)
7. Dossot, D., DÉmic, J.: Mule in Action. Manning Publications (2010)
8. Etzion, O., Niblett, P.: Event Processing in Action. Manning Publications (agosto 2010)
9. Inc., E.: Esper 4.0.0 - reference documentation (2009), http://esper.codehaus.org/esper/documentation/documentation.html
10. Ismail, M.N., Ismail, M.T.: Framework of Intrusion Detection System via Snort Application on Campus Network Environment. In: International Conference on Future Computer and Communication, 2009. ICFCC 2009. pp. 455–459. IEEE (abril 2009)
11. Kou, X., Wen, Q.: Intrusion Detection Model Based on Android. In: 2011 4th IEEE International Conference on Broadband Network and Multimedia Technology (ICBNMT). pp. 624–628. IEEE (octubre 2011)
12. Michelson, B.M.: Event-driven architecture overview: Event-Driven SOA Is Just Part of the EDA Story (febrero 2006), http://dx.doi.org/10.1571/bda2-2-06cc
13. Rademakers, T., Dirksen, J.: Open Source ESBs in Action. Manning Publications (2009)
14. Sward, R.E., Boleng, J.: Service-Oriented Architecture (SOA) concepts and Implementations. In: Proceedings of the 2011 ACM annual international conference on Special interest group on the ada programming language. pp. 3–4. New York, NY, USA (2011)
15. Zappia, I., Paganelli, F., Parlanti, D.: A lightweight and Extensible Complex Event Processing System for Sense and Respond Applications. Expert Systems with Applications
16. Zhai, J., Xie, Y.: Researh on Network Intrusion Prevention System Based on Snort. In: 2011 6th International Forum on Strategic Technology (IFOST). vol. 2, pp. 1133–1136. IEEE (agosto 2011)

Sesion 6

Procesos de Negocios II

Chair: *Dr. Juan Manuel Vara*

Sesion 6: Procesos de Negocios II
Chair: Dr. Juan Manuel Vara

Cristina Cabanillas, Adela Del-Río-Ortega, Manuel Resinas and Antonio Ruiz-Cortés. *RAL Solver: a Tool to Facilitate Resource Management in Business Process Models*.

Cristina Cabanillas, Manuel Resinas, and Antonio Ruiz-Cortés. *Defining and Analysing Resource Assignments in Business Processes with RAL*.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): Actas de las "VIII Jornadas de Ciencia e Ingeniería de Servicios (JCIS'2012)", Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería

RAL Solver: a Tool to Facilitate Resource Management in Business Process Models*

Cristina Cabanillas, Adela del-Río-Ortega, Manuel Resinas, and Antonio Ruiz-Cortés

Universidad de Sevilla, Spain
{cristinacabanillas, adeladelrio, resinas, aruiz}@us.es

Abstract. Business process (BP) modelling notations tend to stray their attention from resource management, unlike other aspects such as control flow or even data flow. On the one hand, the languages they offer to assign resources to BP activities are usually either little expressive, or hard to use for non-technical users. On the other hand, they barely care about the subsequent analysis of resource assignments, which would enable the detection of problems and/or inefficiency in the use of the resources available in a company. We present RAL Solver, a tool that addresses the two aforementioned issues, and thus: (i) allows the specification of assignments of resources to BP activities in a reasonably simple way; and (ii) provides capabilities to automatically analyse resource assignments at design time, which allows extracting information from BP models, and detecting inconsistencies and assignment conflicts.

1 Motivation

Business processes (BPs) are often analysed in terms of control flow, temporal constraints, data and resources. From all of these aspects, resources have received much less attention than other aspects, specially control flow. However, the participation of people in BPs guides the execution of BPs, so human resources should be considered when designing and modelling the BPs used in an organization. In the following we present a tool that is the result of previous work we have carried out addressing different problems on human resource management (*resource management* for short) in BP models [1, 2].

In [1] we dealt with the assignment of resources to the activities of a BP model, aiming at easing and improving the way resources can be associated to the process activities. Some approaches pursuing a similar purpose had been introduced in recent years [3, 4], but they were in general either too complex to be used by technically unskilled people, or not expressive enough to provide powerful resource management in workflows (WFs) and BPs. In that work we introduced RAL (Resource Assignment Language), a DSL (Domain Specific

* This work has been partially supported by the European Commission (FEDER), Spanish Government under project SETI (TIN2009-07366); and projects THEOS (TIC-5906) and ISABEL (TIC-2533) funded by the Andalusian Local Government.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 175-178, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

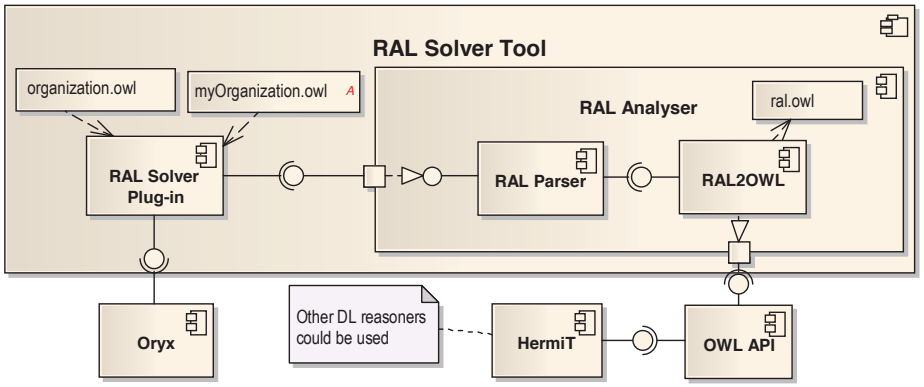


Fig. 1: RAL Solver's Architecture

Language) specifically developed to express resource assignments in BP activities. RAL expressions cover from simple assignments of activities to specific individuals of the company, to complex assignments containing access-control constraints (e.g. Segregation of Duty -SoD-) between activities, as well as compound expressions. As can be seen in the following examples, its syntax close to natural language increases its understandability:

```
RAL 1: IS Samuel
RAL 2: NOT (IS PERSON WHO DID ACTIVITY CreateResolutionProposal)
RAL 3: (HAS ROLE DocumentWriter) OR (HAS ROLE DocumentSigner)
```

In [2] we presented the design-time analysis capabilities provided by RAL. In particular, information such as (i) the potential performers of each BP activity, i.e., the set of people that meet the resource-related constraints imposed in the model (by means of assignment expressions); or (ii) the potential set of activities each person of an organization can be allocated at run time; can be automatically obtained from *RAL-aware BP models*. This has been managed by providing formal semantics to RAL based on description logics (DLs), which allows us to use DL reasoners to automatically infer information about resource management from RAL expressions. RAL Solver has been developed both to prove the use of RAL expressions in process models, and the benefits of its DL-based semantics to analyse RAL-aware BP models.

2 RAL Solver. Definition and Architecture

As explained in [1], RAL can be easily used with BPMN 2.0 [5], but it could also be integrated into other WF modelling notations, provided that they offer some mechanism to include resource assignments in a process model. Regarding BPMN, class *ResourceAssignmentExpression* allows to specify free resource assignments for an activity. These expressions are specifically configured in class *FormalExpression*, which contains two main attributes. For us, attribute *language* takes value "RAL" and the RAL expression can be set in attribute *body*.

RAL Solver has been implemented as a plug-in for Oryx[1], a powerful BP modelling infrastructure that emerged as an Academic Initiative in the HPI in Potsdam. What we do is to add RAL assignments to the activities of the BPMN diagrams modelled with Oryx, making use of the aforementioned BPMN features. Besides, the organizational structure of the company that uses the plug-in has to be modelled as an ontology (*myOrganization.owl*) on the basis of an ontology provided by RAL (*organization.owl*), which is implicitly included in the tool. This can now be done with an ontology editor such as Protégé, but we intend to automate this task so that *myOrganization.owl* is automatically generated from the organizational model of the company.

After receiving these entries the *RAL Analyser* component is launched. This component could also be integrated and used in other platforms. Within it, *RAL Parser* takes the RAL expressions of the BP activities and parses them in order to check their syntax and prepare them for the next step. Then, the RAL expressions are automatically mapped into an OWL ontology necessary to later infer knowledge related to how resources are managed in the process. File *ral.owl* is thus created by component *RAL2OWL*. Finally, *OWL API* is used to check the ontologies, and a DL reasoner can be used to compose and execute analysis operations over the resource assignments (in our case *HermiT*).

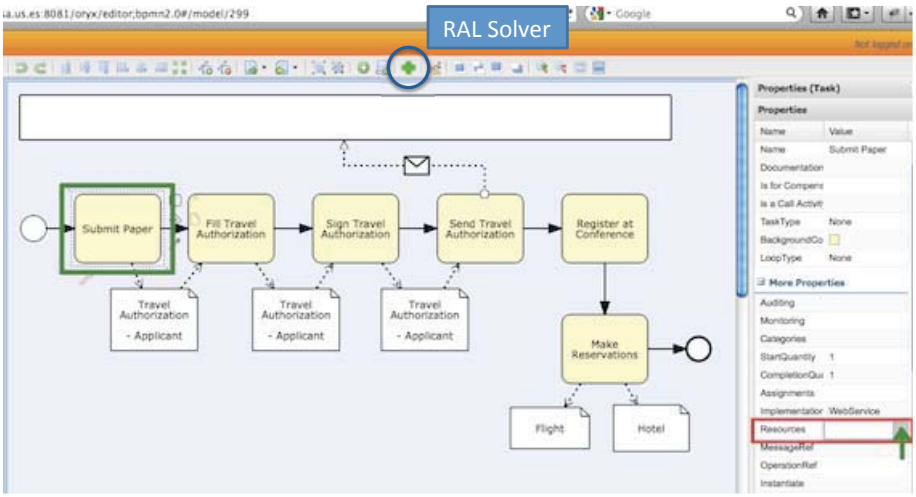
3 How to Use the Tool

The current version of RAL Solver allows the design-time analysis of RAL-aware BPMN models to calculate the set of candidate persons to perform every activity of a process. Run-time support is currently being incorporated as well.

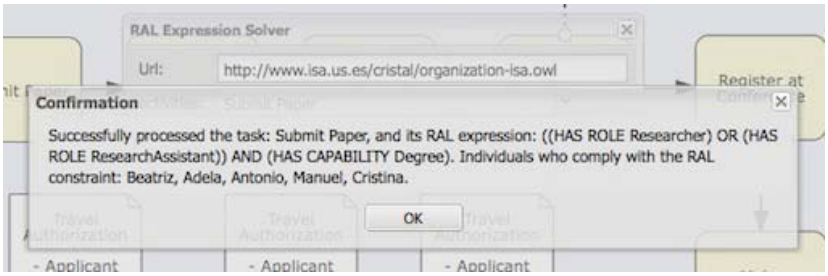
Our distribution of Oryx including the plug-in can be accessed by opening `http://labs.isa.us.es:8081/backend/poem/repository` in Firefox. As shown in Figure 2a, Oryx lets modify certain properties of the BP activities, among them the resource assignments. In the emerging window the RAL expression must be written in field *ResourceAssignmentExpression* and, optionally, we can set field *Language* to *RAL* to give further information about the language used. Afterwards, when we run RAL Solver we must indicate the *myOrganization.owl* file we are going to use as organizational structure, and select the BP activity for which we want to know the potential performers. The result of the analysis is returned in a new window as illustrated in Figure 2b. With this analysis operation we get to know whether, given a collection of resource assignments, all the activities of the process can be performed at run time or, on the contrary, certain constraints set produce the absence of potential performers for any task. To implement more analysis operations we would make use of other operations provided by the DL reasoner.

Further information about RAL, its analysis features, and how to use RAL Solver can be found at `http://www.isa.us.es/cristal`. The tool is available on demand via email.

[1] `http://bpt.hpi.uni-potsdam.de/Oryx/`



(a) Editor appearance



(b) Result of the analysis

Fig. 2: Integration of RAL Solver into Oryx

References

1. C. Cabanillas, M. Resinas, and A. Ruiz-Cortés, "RAL: A High-Level User-Oriented Resource Assignment Language for Business Processes," in *Business Process Management Workshops (BPD'11)*, pp. 50–61, 2012.
2. C. Cabanillas, M. Resinas, and A. Ruiz-Cortés, "Defining and Analysing Resource Assignments in Business Processes with RAL," in *ICSOC*, pp. 477–486, 2011.
3. A. Awad, A. Grosskopf, A. Meyer, and M. Weske, "Enabling Resource Assignment Constraints in BPMN," tech. rep., BPT, 2009.
4. C. Wolter, P. Miseldine, and C. Meinel, "Verification of Business Process Entailment Constraints Using SPIN," in *Engineering Secure Software and Systems*, vol. 5429, pp. 1–15, 2009.
5. "BPMN 2.0," recommendation, OMG, 2011.

Summary of "Defining and Analysing Resource Assignments in Business Processes with RAL" [1]

Cristina Cabanillas, Manuel Resinas, and Antonio Ruiz-Cortés

Universidad de Sevilla, Spain
{cristinacabanillas, resinas, aruiz}@us.es

Summary of the Contribution

Business processes (BPs) are often analysed in terms of control flow, temporal constraints, data and resources. From all of these aspects, resources have received much less attention than other aspects, specially control flow. Even the standard BP modelling notation (BPMN) does not provide concrete definitions for the resource-related concepts [2]. However, the participation of people in BPs is of utmost importance, both to supervise the execution of automatic activities and to carry out software-aided and/or manual tasks. Therefore, they should be considered when designing and modelling the BPs used in an organization.

In this paper we face human-resource management (*resource management* for short) in BP models. Firstly, we deal with the assignment of resources to the activities of a BP model, aiming at easing and improving the way resources can be associated with BP activities. Some approaches addressing a similar purpose have been introduced in the last years [3–5], but they are in general either too complex to be used by technically unskilled people, or not expressive enough to provide powerful resource management in workflows (WFs) and BPs.

Another substantial shortage in many existing approaches related to resource management in WFs and BPs (e.g. in the default resource management mechanism provided by BPMN 2.0), is that they do not bridge the gap between organizational models and BP models, which means they do not actually relate both elements, thus treating them separately. Our second goal in this work is to come up with a solution that narrows this gap, at the same time as it allows us to analyse the resource assignments associated to the BP activities. With such a solution, organizations can benefit from the *automation* of work in different directions, to be named:

- *The inference of interesting information*, such as: (i) the potential performers of each BP activity, i.e., the set of people that meet the resource-related constraints imposed in the model (by means of assignment expressions associated to the BP activities); or (ii) the potential set of activities each person of an organization can be allocated at run time. This kind of information may be beneficial for an organization in several ways. For instance, in the previous case: the former benefits the person in charge of resource allocation, since it increases the information available to allocate tasks to resources

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 179-180, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

when the BP is executed; and the latter provides an employee-oriented vision, informing about the possible workload of each employee and, hence, allowing reacting in time to avoid having people overburderned with work.
- *The detection of inconsistencies* between the resource assignments associated to activities of a BP model and the structure of the organization where it is used, e.g. non-existent roles or persons.

The contribution of this paper is, hence, twofold:

1. On the one hand, we have developed RAL (Resource Assignment Language), a DSL (Domain Specific Language) to express resource assignments in BP activities. It is based on an organizational metamodel proposed by Russell et al. [6], which considers the company structured according to persons, organizational units, roles and a hierarchy of positions. RAL was previously introduced in [7]. Its expressiveness has been tested using the group of *creation patterns* of the well-known *Workflow Resource Patterns* [6].
2. On the other hand, we provide a semantic mapping of RAL expressions into description logics (DLs). This allows us to use DL reasoners to automatically infer information about resource management from RAL assignments, and detect assignment inconsistencies, as explained above.

RAL Solver has been developed as a plugin for Oryx [8], to prove the use of RAL expressions in BP models and the benefits of its DL-based semantics. More information about the language and its analysis features can be found at `http://www.isa.us.es/cristal`.

References

1. C. Cabanillas, M. Resinas, and A. Ruiz-Cortés, "Defining and Analysing Resource Assignments in Business Processes with RAL," in *ICSOC*, vol. 7084, pp. 477–486, 2011.
2. "BPMN 2.0," recommendation, OMG, 2011.
3. A. Awad, A. Grosskopf, A. Meyer, and M. Weske, "Enabling Resource Assignment Constraints in BPMN," tech. rep., BPT, 2009.
4. C. Wolter, P. Miseldine, and C. Meinel, "Verification of Business Process Entailment Constraints Using SPIN," in *Engineering Secure Software and Systems*, vol. 5429 of *Lecture Notes in Computer Science*, pp. 1–15, Springer Berlin / Heidelberg, 2009.
5. M. Strembeck and J. Mendling, "Modeling process-related RBAC models with extended UML activity models," *Inf. Softw. Technol.*, vol. 53, pp. 456–483, 2011.
6. N. Russell, W. M. P. van der Aalst, A. H. M. ter Hofstede, and D. Edmond, "Workflow resource patterns: Identification, representation and tool support," in *CAiSE*, pp. 216–232, 2005.
7. C. Cabanillas, M. Resinas, and A. Ruiz-Cortés, "RAL: A High-Level User-Oriented Resource Assignment Language for Business Processes," in *BPM Workshops (BPD'11)*, pp. 50–61, 2011.
8. G. Decker, H. Overdick, and M. Weske, "Oryx - an open modeling platform for the BPM community," in *Business Process Management (BPM)*, pp. 382–385, 2008.

Charla Invitada

Crossing the Software Education Chasm using Software-as-a-Service and Cloud Computing

Armando Fox

A. Ruíz, L. Iribarne (Eds.): Actas de las "*XVII Jornadas de Ingeniería del Software y Bases de Datos (JISBD'2012)*", Jornadas SISTEDES'2012, Almería 17-19 sept. 2012, Universidad de Almería.

Crossing the Software Education Chasm using Software-as-a-Service and Cloud Computing

Prof. Armando Fox
Computer Science Division, University of California, Berkeley
fox@cs.berkeley.edu

Via the remarkable alignment of cloud computing, software as a service (SaaS), and Agile development, the future of software has been revolutionized in a way that also allows us to teach it more effectively. Over the past 3 years we have been reinventing UC Berkeley's undergraduate software engineering course to cross the long-standing chasm between what many academic courses have traditionally offered and the skills that software employers expect in new hires: enhancing legacy code, working with nontechnical customers, and effective testing. In our course, "two-pizza teams" of 4 to 6 students create a prototype application specified by real customers (primarily nonprofit organizations) and deploy it on the public cloud using the Rails framework and Agile techniques. Students employ user stories and behavior-driven design to reach agreement with the customer and test-driven development to reduce mistakes. During four 2-week iterations, they continuously refine the prototype based on customer feedback, experiencing the entire software lifecycle–requirements gathering, testing, development, deployment, and enhancement–multiple times during a 14-week semester. Because of Rails' first-rate tools for testing and code quality, students learn by doing rather than listening, and instructors can concretely measure student progress. We have also successfully repurposed those same tools to support nontrivial machine grading of complete programming assignments, allowing us to scale the on-campus course from 35 to 115 students and offer a Massively Open Online Course (MOOC) to over 50,000 students. Indeed, to support instructors interested in adopting our techniques in their classes, we provide not only an inexpensive textbook and prerecorded video lectures to complement the curriculum, but also a set of questions and programming assignments that includes free autograding. Our experience has been that students love the course because they learn real-world skills while working with a real customer, instructors love it because students actually practice what they learn rather than listening to lecture and then coding the way they always have, and employers love it because students acquire vital skills missing from previous software engineering courses.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 183, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

A. Ruíz, L. Iribarne (Eds.): Actas de las "*XVII Jornadas de Ingeniería del Software y Bases de Datos (JISBD'2012)*", Jornadas SISTEDES'2012, Almería 17-19 sept. 2012, Universidad de Almería.

Sistedes 2012
#sistedes2012 Almería

JISBD PROLE JCIS

ACG
Applied Computing Group
Ref. TIC-211, Junta de Andalucía
University of Almería, Spain
Ctra. Sacramento s/n, 04120 Almería

FUNDACIÓN MEDITERRÁNEA
EMPRESA - UNIVERSIDAD DE ALMERIA

Colaboradores

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y COMPETITIVIDAD

JUNTA DE ANDALUCIA
CONSEJERÍA DE ECONOMÍA, INNOVACIÓN Y CIENCIA

DIPUTACIÓN DE ALMERÍA

AYUNTAMIENTO DE ALMERÍA

ALBORADA Ingenieros I+D+I

Escuela Superior de Ingeniería
UNIVERSIDAD DE ALMERÍA

UNIVERSIDAD DE ALMERÍA
Vic. de Tecnologías de la Información y de la Comunicación
Vic. de Investigación, Desarrollo e Innovación
Unidad de Enseñanza Virtual del VTIC (EVA/TIC)
Departamento de Lenguajes y Computación

UNIVERSIDAD DE ALMERÍA
Plan Propio de Investigación

cajamar

PITA Parque Científico-Tecnológico de Almería

IEEE computer society
IEEE Computer Society – Spanish Chapter

ices Ingeniería, Consultoría y Estrategia de Sistemas

almerimatik i+d+i

http://sistedes2012.ual.es