# ICWS 2006
# Message from the General Chairs

Welcome to the 2006 IEEE International Conference on Web Services (ICWS 2006)! ICWS 2006 is now in its fourth anniversary. The long-term goal of ICWS is to establish a reputable and respectable conference for the international community of researchers and practitioners to exchange information regarding advancements in the state of the art and practice of Web services, and to identify the emerging research topics and define the future of Web services computing. ICWS 2006 is sponsored by IEEE Computer Society Technical Committee on Services Computing (TC-SVC) and is co-located with the 2006 IEEE International Conference on Services Computing (SCC 2006), as well as the 30th Annual International Computer Software and Applications Conference (COMPSAC 2006).

Web services are Internet-based application components published using standard interface description languages and become universally available via uniform communication protocols. As in the fourth year, the program of ICWS 2006 continues to feature research papers with a wide range of topics, focusing on various aspects of IT services such as Web services specifications and enhancements, Web services discovery and integration, Web services security, Web services Modeling, Web services-based Software Engineering, Web services-based applications and solutions, and semantics in Web services.

As the most prestigious academic conference in the field of Web services, the first International Conference on Web Services, ICWS 2003 was held at the Monte Carlo Resort in Las Vegas, Nevada, June 23 - 26, 2003, attracting hundreds of participants from 25 countries. ICWS 2004 was held at Westin Horton Plaza in San Diego, California, July 6-9, 2004, attracting about 250 registered participants from 22 countries and regions. The joint ICWS 2005 and SCC 2005 attracted more than 300 registered participants. ICWS 2003, ICWS 2004, and ICWS 2005 have proven to be excellent catalysts for research and collaboration, and we fully expect that this coming ICWS 2006 conference will continue this trend.

Many people have worked very hard to make this conference possible. We would like to thank all who have helped in making ICWS 2006 a success. The Program Committee members and referees all deserve credit for producing the excellent final program that resulted from the diligent review of the submissions. Special thanks go to the program chairs and all the other organizing committee members. We would also like to appreciate the Services Computing Professional Interest Community (PIC) at IBM Research and IEEE IT Professional Magazine. Clearly, there are still many unresolved research issues in regard to Web services computing as a very young field. We hope you enjoy the conference and plan to contribute to the future ICWS events as authors, speakers, panelists, and volunteer as conference organizers. Finally, ICWS belongs to you, the audience.

**Frank Leymann, Ph.D.**
*General Co-Chair of IEEE ICWS 2006*
*University of Stuttgart, Germany*

**Liang-Jie (LJ) Zhang, Ph.D.**
*General Co-Chair of IEEE ICWS 2006*
*IBM T.J. Watson Research, USA*

# ICWS 2006
# Message from the Program Chairs

Welcome to ICWS 2006! Over the past four years, ICWS has grown to be a prime international forum for both researchers and practitioners to exchange the latest fundamental advances in the state of the art and practice of Web services and to envision the future of Web services.

This year's conference attracted more than 300 submissions from all over the world. Except for a very small number of papers that received 2 reviews, each paper was reviewed by 3 or 4 program committee members. After initial review and follow-up discussions, the program committee selected 48 articles as research papers.

The research track is organized into 16 sessions covering a wide range of topics including: Web services matchmaking and discovery, QoS issues for Web services, Web services composition and integration, semantics in Web services, Web services specifications and enhancement, modeling and benchmarking, SOA and services-oriented software engineering, business process management, and services-based mobile computing. In addition to the research track, we offer two new programs: one is an application services and industry track, focusing on services realizations and industrial practices, and the other is a works-in-progress track, focusing on on-going services-oriented research and practice. Furthermore, the joint ICWS and SCC programs feature 3 keynotes by leaders in Web services research and development, 6 tutorials, and 4 panels.

We thank all the authors who submitted top-quality papers to the conference. The decision making was extremely difficult but the results are exceptional. We are very grateful to the outstanding international cast of 96 program committee members and the many external reviewers. Their timely completion of more than 1000 reviews and active participation in discussions was critical in making the review and selection process possible. We also thank the ICWS steering committee, our General Co-Chairs Frank Leymann and Liang-Jie (LJ) Zhang, and the ICWS 2006 Organizing Committee for their help in putting together such an exciting conference. Finally, we thank all of you who come to the conference. We hope you find the conference both stimulating and enjoyable!

**Ephraim Feig, Ph.D.**
*Program Co-Chair of IEEE ICWS 2006*
*Motorola, USA*

**Anup Kumar, Ph.D.**
*Program Co-Chair of IEEE ICWS 2006*
*University of Louisville, USA*

**Jia Zhang, Ph.D.**
*Program Vice-Chair of IEEE ICWS 2006*
*Northern Illinois University, USA*

# ICWS 2006
# Message from the Application Services and Industry Track Chair

Web services have been evolving rapidly in recent years and become a fertile research area for both academia and industry. The advance of Web services is strongly influenced by the need in real applications, as it extends many principles and practices of Web to enable dynamic program-to-program interactions. The adoption of Web service is accelerated by the strong industry standardization efforts which define the fundamental infrastructure, e.g. SOAP, WSDL, etc., as well as critical service protocols, e.g. WS-Addressing, WS-Security, UDDI, etc., for Web service invocation and enablement.

The Application Services and Industry Track of the 4th IEEE International Conference on Web Services (ICWS 2006) is intended to provide a forum for researchers and practitioners from industry and academia to exchange ideas, share experiences, and foster new initiatives that can advance Web services to the next level. As the industry is moving towards a service-oriented architecture (SOA) and a more dynamic event-driven architecture (EDA), there is an increasing need to motivate new paradigms and approaches, in order to address technical challenges arising from new applications.

The Application Services and Industry Track of ICWS 2006 consists of 17 sessions. It provides a balanced coverage encompassing various aspects of Web service technologies. After a peer review process, 51 papers were accepted for the proceedings of ICWS 2006 and for presentation at the conference.

In addition to the requirement of scientific rigor, the Application Services and Industry Track paper review process placed an emphasis on the significance of the contributions to address real problems in application services and industry applications. Many people contributed to this process.  And we would like to thank all of them for their hard work and dedication which laid the foundation for the success of the Application Services and Industry Track at ICWS 2006.

Moreover, I would like to thank all paper authors who submitted their papers to ICWS 2006. It is their contribution that made ICWS 2006 a premier conference in Web services, and upon which Application Services and Industry Track at ICWS 2006 can provide a meeting place for both industry and academia in their pursuit of new advances in Web services.

**Wu Chou, Ph.D.**
*Application Services and Industry Track Chair of IEEE ICWS 2006*
*Avaya Labs Research, Avaya, USA*

# ICWS 2006
# Technical Steering Committee

**Carl K Chang**, Iowa State University, USA
**Ephraim Feig**, Motorola, USA
**Hemant Jain**, University of Wisconsin- Milwaukee , USA
**Frank Leymann**, University of Stuttgart, Germany
**Calton Pu**, Georgia Tech, USA
**Ming-Chien Shan**, Hewlett-Packard, USA
**Jeffrey Tsai**, University of Illinois at Chicago, USA
**Liang-Jie Zhang (Chair)**, IBM T.J. Watson Research Center, USA

# ICWS 2006 Conference Officers

## General Chairs

**Frank Leymann, Ph.D.**
Professor, University of Stuttgart, Germany

**Liang-Jie Zhang, Ph.D.**
Research Staff Member, IBM T.J. Watson Research Center, USA

## Program Committee Chairs

**Ephraim Feig, Ph.D.**
Motorola, USA

**Anup Kumar, Ph.D.**
Professor, University of Louisville, USA

## Program Committee Vice Chair

**Jia Zhang, Ph.D.**
Assistant Professor, Northern Illinois University, USA

## Workshop Chairs

**Malu G. Castellanos, Ph.D.**
HP Laboratories, USA

**Jian Yang, Ph.D.**
Associate Professor, Macquaire University, Australia

## Application Services and Industry Track Chair

**Wu Chou, Ph.D.**
Avaya Labs Research, Avaya, USA

## IEEE SOA Industry Summit Chairs

**Ali Arsanjani, Ph.D.**
Chief Architect, SOA and Web Services Center of Excellence
IBM Global Services, USA

**Tony Shan**
Lead Systems Architect, Wachoiva Bank, USA

## Tutorial Chairs

**Schahram Dustdar, Ph.D.**
Professor, Vienna University of Technology, Austria

**Andreas Wombacher, Ph.D.**
PostDoc Fellow, University of Twente, The Netherlands

## Job Fair Chairs

**Sandeep Purao, Ph.D.**
Associate Professor, Pennsylvania State University, USA

**Sriram Anand, Ph.D.**
Principal Researcher
Infosys Technologies, India

## Panel Chairs

**Frank Ferrante**
Chair, IEEE Computer Society Communities (EPS), USA

**Ling Liu, Ph.D.**
Associate Professor, Georgia Institute of Technology, USA

**Hsing Kenny Cheng, Ph.D.**
Associate Professor, University of Florida, USA

## Work-in-Progress Chair

**Graciela Gonzalez, Ph.D.**
Visiting Assistant Professor, Arizona State University, USA

## Publicity Chairs

**Jeff Voas, Ph.D.**
Director, Systems Assurance
Science Applications International Corporation, SAIC, USA

**Elena Ferrari, Ph.D.**
Professor, University of Insubria at Como, Italy

**Steve Miller, Ph.D.**
Professor, Dean of the School of Information Systems
Singapore Management University, Singapore

## Publication Chair

**Haifei Li, Ph.D.**
Assistant Professor, Union University, USA

## Registration Chair

**Huimin Chen, Ph.D.**
Assistant Professor, University of New Orleans, USA

## IEEE APSCC 2006 Liaison

**Hai Jin, Ph.D.**
Professor and Dean, School of Computer
Huazhong University of Science and Technology, China

## IEEE Services Computing Contest Chairs

**Zhixiong Chen, Ph.D.**
Associate Professor, Mercy College, USA

**Charles A. Shoniregun, Ph.D.**
University of East London, UK

**Yuan-Chwen You, Ph.D.**
Founder, Creative Entrepreneurship Consulting, Inc., Taiwan

# ICWS 2006
# Program Committee

**Rama Akkiraju**, IBM T.J. Watson Research, USA
**Grigoris Antoniou**, University of Crete/Institute of Computer Science FORTH, Greece
**Mikio Aoyama**, Nanzan University, Japan
**Ali Arsanjani**, IBM Global Services, USA
**Malcolm Atkinson**, University of Edinburgh, UK
**Boualem Benatallah**, University of New South Wales, Australia
**Elisa Bertino**, Purdue University, USA
**Ken Birman**, Cornell University, USA
**Athman Bouguettaya**, Virginia Tech., USA
**Paul Buhler**, College of Charleston, Charleston, SC USA
**Christoph Bussler**, Cisco Systems, USA
**Jorge Cardoso**, University of Madeira, Portugal
**Malu G. Castellanos**, HP Labs, USA
**Jeane Chen,** Kintera, USA
**Ying Chen**, IBM China Research Lab, China
**Dickson K.W. Chiu**, Dickson Computer Systems, Hong Kong, China
**Wu Chou**, Avaya Labs Research, Avaya, USA
**Alok Choudhary**, Northwestern University, USA
**Vassilis Christophides**, University of Crete/Institute of Computer Science FORTH, Greece
**Umeshwar Dayal**, HP Labs, USA
**Stefan Dessloch**, Kaiserslautern University of Technology, Germany
**Asuman Dogac**, Middle East Technical University, Turkey
**Jiang Du**, Queensland University of Technology, Australia
**Schahram Dustdar**, Vienna University of Technology, Austria
**Phillip Ein-Dor**, Tel-Aviv University, Israel
**Vadim Ermolayev**, Zaporozhye State University, Ukraine
**Ben Falchuk**, Telcordia Technologies, Inc., USA
**Dieter Fensel**, DERI, Ireland & University of Innsbruck, Austria
**Elena Ferrari**, Politiche e dell'Informazione, University of Insubria at Como, Italy
**Marcus Fontoura**, Yahoo Research, USA
**Piero Fraternali**, Politecnico di Milano, Italy
**Casey K. Fung**, Boeing Phantom Works, USA
**Martin Gaedke**, University of Karlsruhe, Germany
**George M. Galambos**, IBM Canada
**Shahram Ghandeharizadeh**, University of Southern California, USA
**Graciela Gonzalez**, Arizona State University, USA
**Vijay K. Gurbani**, Lucent Technologies/Bell Labs Innovations, USA
**Abdelsalam, Sumi Helal**, University of Florida, USA
**Alan R. Hevner**, University of South Florida, USA
**Michael N. Huhns**, University of South Carolina, USA
**Rick Hull**, Lucent Technologies, Bell Labs, USA

**Patrick C. K. Hung**, University of Ontario Institute of Technology, Canada

**Varghese S. Jacob**, University of Texas at Dallas, USA

**Hemant Jain**, University of Wisconsin - Milwaukee, USA

**Ralph Johnson**, University of Illinois at Urbana-Champaign, USA

**Anupam Joshi**, University of Maryland, Baltimore County, USA

**Roger, Buzz King**, University of Colorado at Boulder, USA

**Hiroyuki Kitagawa**, University of Tsukuba, Japan

**Domenico Laforenza**, Information Science and Technologies Institute (ISTI), Italy

**Herman Lam**, University of Florida, USA

**Konstantin Läufer**, Loyola University Chicago, USA

**M. Lenzerini**, DIS (Dipartimento di Informatica e Sistemistica), Italy

**Frank Leymann**, University of Stuttgart, Germany

**Haifei Li**, Union University, USA

**Wei Li**, Institute of Computing Technology, Chinese Academy of Sciences, China

**Geng Lin**, Cicso Systems, USA

**Ling Liu**, Georgia Institute of Technology, USA

**Bertram Ludaescher**, University of California at Davis, USA

**J.P. Martin-Flatin**, UQAM, Canada

**XS (Xin Sheng) Mao**, IBM China Software Development Laboratory (CSDL), China

**Hiroshi Maruyama**, IBM Research, Japan

**Carolyn McGregor**, University of Western Sydney, Australia

**Dennis McLeod**, Univeristy of Southern California, USA

**Hong Mei**, Beijing University, China

**Xiannong Meng**, Bucknell University, USA

**Dejan S. Milojicic**, HP Laboratories, USA

**Simanta Mitra**, Iowa State University, USA

**Tadao Murata**, University of Illinois at Chicago, USA

**Jeff Offutt**, George Mason University, USA

**Massimo Paolucci**, DoCoMo Euro-Labs, Germany

**Thomas E. Potok**, Oak Ridge National Lab (ORNL), USA

**Thierry Priol**, Inria, France

**Calton Pu**, Georgia Tech, USA

**Norbert Ritter**, University of Hamburg, Germany

**Dumitru Roman**, University of Innsbruck / DERI Innsbruck, Austria

**Steve Ross-Talbot**, Enigmatec Corporation, Germany

**Atul Sajjanhar**, Deakin University, Australia

**Vipul Shah**, Tata Consultancy Services America, USA

**Amit Sheth**, University of Georgia and Semagix Inc., USA

**Simon Shim**, San Jose State University, USA

**Munindar P. Singh**, North Carolina State University, USA

**Il-Yeol Song**, Drexel University, USA

**Judith Stafford**, Tufts University, USA

**Rudi Studer**, University Karlsruhe, Germany

**Jianwen Su**, University of California at Santa Barbara, USA

**Stanley Su**, University of Florida, USA

**Katia Sycara**, Carnegie-Mellon University, USA

# ICWS 2006
# External Reviewers

Suhir Agarwal
Bugrahan Akcay
Jiyuan An
Anupriya Ankolekar
Marco Brambilla
Amit Chopra
Sara Comai
Xiaohui Cui
Jiangbo Dang
Nirmit Desai
Robert Dew
Christoph Dorn
Robin Doss
Eduard Dragut
Ru Fang
Xing Fang
Cristina Feier
Subhankar Ghosh
Ozgur Gulderen
Yanan Hao
Andreas Heil
Jingyu Hou
Gang Huang
Jingshan Huang
Ruo Bo Huang
Mathieu Jan
Jean-Marc Jézéquel
Wenpin Jiao
Yu Jiao
Yildiray Kabak
Uwe Keller

Mick Kerrigan
Reto Krummenacher
Gokce Banu Laleci
Holger Lausen
Benito Mendoza
Adrian Mocan
Li Li
Ming Li
Ying Li
Alice (Ying) Liu
Fang Liu
Shuang Liu
Xuanzhe Liu
Xumin Liu
Zhiyi Ma
Zaki Malik
Zsolt Nemeth
Alper Okcan
Mehmet Olduz
Umut Orhan
Robert Patton
Vishnu Pendyala
Andrea Perego
Christian Pérez
Trivikram Phatak
Jean-Marc Pierson
Christian Platzer
Dumitru Roman
Florian Rosenberg
Jie Qiu
Daniel Schall

James Scicluna
Tayfun Sen
Jyothi Shettigar
Adina Sirbu
Andrzej Skowron
Nenad Stojanovic
Lianshan Sun
Xi Sun
Ibrahim Tasyurt
George K. Thiruvathukal
Ioan Toma
Martin Treiber
Yathiraj Udupi
Ning Wu
Yang Xiang
Bing Xie
Bo Yang
Jie Yang
Xu Yang
Mark L Yi
Chun Ying
Qi Yu
Mustafa Yuksel
Michal Zaremba
Wei Zhang;
Wen Zhao
George Zheng
Minghui Zhou
Wei Zhou

# PWSSec: Process for Web Services Security

C.Gutiérrez [1], E. Fernández-Medina[2] and M. Piattini[2]
*(1) STL, Madrid (SPAIN), carlos.gutierrez@stl.es*
*(2) Alarcos Research Group. Universidad de Castilla-La Mancha.*
*Paseo de la Universidad 4, 13071, Ciudad Real. (SPAIN). Tel: 34 926 29 53 00*
*{Eduardo.FdezMedina, Mario.Piattini}@uclm.es*

## Abstract

*In the last few years, the field of Web Services (WS) security has evolved rapidly producing an impressive number of WS-based security standards. This fact has caused that organizations are still reticent about adopting technologies based on this paradigm due to the learning curve necessary to integrate security into their practical deployments. In this paper, we present PWSSec (Process for Web Services Security) as a process that enables the integration of a set of specific stages into the traditional phases of WS-based systems development providing them with security. PWSSec is composed of three stages, WSSecReq (Web Services Security Requirements), WSSecArch (Web Services Security Architecture) and WSSecTech (Web Services Security Technologies) that allow the specification of WS-specific security requirements, the definition of the WS-based security architecture and the identification of the security standards that the security architecture must deploy, respectively.*

## 1. Introduction

The open Internet nature is promoting the development of Web Services (WS) as a paradigm that enables complex business workflow integration scenarios and provides the so-demanded and so-called hyper-connectivity inter- and intra-enterprises [1]. This standard-based quality-centered paradigm is evolving rapidly due to its capability of handling and addressing the heterogeneous system integration issue. In fact, according to the most recent reports from IDC, over the next years, the market for WS-based solutions will grow steadily reaching $11 billion in 2008 [2]. This growth has caused that in the last few years a lot of initiatives have been developed (apart from others previously existing). Due to this fact, an enormous quantity WS-standards is being produced. This diversity, also found in the context of WS security [3] has made us consider its application, from a global perspective, as a very complex and hard process to understand with a very difficult learning curve. At present, there is still a lack of a global approach that offers a methodical development to construct security architectures for WS-based systems. For this purpose, the main objective of this paper is to present the PWSSec (Process for Web Services Security) process.

PWSSec has been created to facilitate and orientate the development of WS-based security systems so that in each one of the traditional stages for the construction of this kind of systems [4], a complementary stage comprising security [5] can be easily integrated. Therefore, this process can be used once the functional architecture of the system has been built or during the stages used to elaborate this architecture. In both cases, the result will be a security architecture formed by a set of coordinated security mechanisms that use the WS security standards to fulfil the system security requirements.

In this paper we will provide a brief overview of the complete process and we will present in a more detailed form the WSSecReq stage.

The rest of the paper is organized as follows. In section 2, the PWSSec process is introduced. In section 3, an in-depth study of the stage related to the specification of security requirements for WS-based systems is presented. In section 4, related research works are stated, and, finally, in section 5 conclusions and issues that need to be developed in the future are stated.

## 2. PWSSec - Process for WS Security

In this section, we will present a general overview of our development process for developing secure WS-based systems.

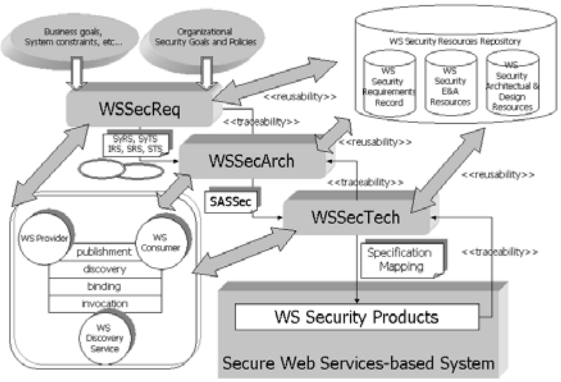
PWSSec follows an iterative and incremental development that facilitates the development and

0-7695-2669-1/06 $20.00 © 2006 IEEE

integration of security into WS-based systems. This process specifies how to define security requirements for WS-based systems describes logical security services-based reference security architecture and explains how to instantiate it to obtain a concrete security architecture based on the current WS security standards. This process can be used either in the development of a new system, since it can be perfectly integrated into the stages in which functional generic services-based [6, 7] and WS-specific [4] development processes are divided, or it can be used for an already designed or developed system based on WS.

The main features of the process presented in this paper are as follows: i) Iterative and incremental process: For each iteration that comprises the development of all stages, a part (increase) of system security is analyzed, characterised and developed; ii) The two basic principles are process traceability and reusability and product interoperability and reusability. iii) Process managed by the elements and basic procedures defined for an Architecture based on WS [8]: the basic actors are the services provider agents, the services consumer agents and the discovery agents whilst the basic processes are WS publishing, discovery, binding and invocation; iv) based on the concept and techniques developed within the scope of Security Requirement Engineering and Risk analysis and management. [9]; v) developed from the concept and techniques that allow us to implement security into software architecture [10-12].

**Figure1. PWSSec stages and main components.**



All the stages of this process must pay attention to the elements and basic processes defined within a SOA (Service-Oriented Architecture) architecture. For instance, as we will see in WSSecReq stage, security requirements must be specified in terms of the agents taking part in SOA architecture: agents consumers and providers of WS and agents acting as a repository of services metadata (WS Discovery Service in Figure 1); and according to the main processes carried out within SOA architecture: processes of publishing, discovery, binding and invocation. Normally, security requirements are erroneously focused on WS providers and invocation process. However, this is a clear mistake because the fact of guaranteeing security in the invocation of a WS is worthless if the processes of discovery or binding (e.g.: trust negotiation interactions [13]) have not been analyzed, designed and developed taking into account security.

Figure 1 shows the stages in which PWSSec process is structured. Each one of the stages defined within PWSSec process describes its inputs, outputs, activities, actors and sometimes, guides, best practices, tools and techniques that complement, improve and facilitate the set of activities developed within these stages.

These stages are briefly described as follows:

1. WSSecReq (Web Services Security Requirements): The main purpose of this stage is to produce a specification (or a part of it) of the security requirements of the target system based on WS. Its input is composed by a specification of the scope that we want to comprise during iteration (e.g.: if we have a definition of the Use Cases available, we can select those that we want to cover and use them as an input for the iteration), the business and security goals defined for the system as well as the part of the organizational security policy that we estimate that may impact on the system design. The output is basically formed by: i) A threat attack tree [14] associated with the WS business and application pattern [4] identified within the analyzed functionality; ii) Every built attack tree's leaf will show a threat [15] that can refined by a set of attack scenarios, defined as misuse cases according to [16, 17], organized into attack profiles [18], and represented according to the Quality if Service UML profile [19]; ii) every misuse case must have related a set of security use cases, according to Donald G. Firesmith [20], that state how the system should respond to the associated misuse case; iii) A formal specification of the security requirements for the scope of the system based on SIREN [21]. These requirements will have been derived after instantiating the WS security requirements templates associated with every security use case.

This stage is supported by two repositories: i) *WS Security E&A Resources*, that contains all the artefacts mentioned above but the security requirements specification; ii) *WS Security Requirements Record* that contains a set of generic security requirements that can be applied to WS-based systems within diverse domains [21].

2. WSSecArch (Web Services Security Architecture), has as its main objective to situate and integrate the security requirements specified in WSSecReq stage through the identification of the appropriate security WS architectural patterns and the security services derived from them. The input of this stage is composed of: i) business goals of the current iteration; ii) organizational security goals and policies taken into account during the current iteration; iii) the set of attack and security scenarios developed in WSSecReq; and iv) the set of security requirements defined in the specifications SyRS, SRS, SyTS, STS, IRS developed in WSSecReq stage. The output is a complete specification of the developed security architecture, called Software Security Architecture Specification (SASSec), indicating: i) how the functional requirements used as input to the stage are integrated into the specifications mentioned above; ii) what security requirements are achieved and how are the allocated in the architecture [22]; and iii) what are the security WS that need to be introduced as security mechanisms. There have been defined four activities that are developed in this stage i) WS Security Architecture Patterns Identification, that defines the mechanisms for the development of the security requirements being analyzed in the current iteration; ii) WS Security Architectural Patterns Integration, that solves the aspects related to security services integration and their interactions with respect to the business services they apply to; iii) Security Architecture Validation that verifies that the attack and security scenarios for the current iteration are covered and identifies possible conflicts of the solution with respect to other functional or quality requirements; iv) WS Security Architecture Specification, that consist of writing the document Security Architecture Specification (SASSec in Figure 1) supported by the use of a set of views [23, 24]. All these four activities are carried out based on a WS security reference architecture we have defined that facilitates the partitioning of the system and the allocation of the security requirements. The core of this WS Security Reference Architecture are the WS Security Kernel (WSSecKern) component, which is responsible for addressing a set of security requirements from the SASSec and the Abstract Security Services attach to them, which are security services that comprises certain set of security requirements types (e.g.: security requirements related to authorization). Due to space limitations, this stage will not be studied in this paper.

3. WSSecTech (Web Services Security Technologies): The main purpose of this stage is to define a set of standards that will implement the Abstract Security Services identified in the previous stage. Its principal input will be the SASSec elaborated then. Output will be a description of the set of standards identified for each Abstract Security Service together with the reasoning framework that made us select it and a security architecture design. The activities carried out in this stage are the following: i) WS-based Security Standards Identification; and ii) Deployment Security Policies Definition.

## 3. WSSecReq – Web Services Security Requirements

In this section, we will describe WSSecReq stage (see Figure 2), of PWSSec process.

### 3.1. Objectives and general considerations

The main objective of this stage is to produce a specification of the security requirements of the WS-based system (or part of it) considered in the current iteration. This stage is developed through a series of activities that are elicitation, analysis, specification and validation of the software security requirements.

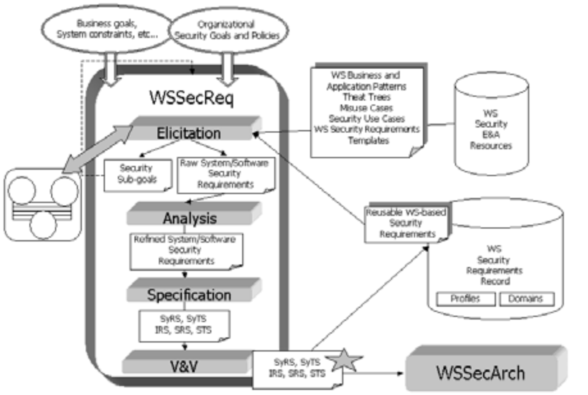

Business goals, System constraints, etc...
Organizational Security Goals and Policies
WSSecReq
Elicitation
Security Sub-goals
Raw System/Software Security Requirements
Analysis
Refined System/Software Security Requirements
Specification
SyRS, SyTS IRS, SRS, STS
V&V
SyRS, SyTS IRS, SRS, STS
WSSecArch
WS Business and Application Patterns Theat Trees Misuse Cases Security Use Cases WS Security Requirements Templates
WS Security E&A Resources
Reusable WS-based Security Requirements
WS Security Requirements Record
Profiles
Domains

**Figure 2. WSSecReq stage: activities and artefacts.**

During the elicitation and analysis of security requirements in WS-based systems, we should take into account the new security scenarios and goals that have appeared such as Single Sign-On solutions, trust domain federation, trust management (including aspects like trust negotiation, trust lifecycle management, policy versioning [13, 25]), protection at the semantic level in which the service must protect information according to its semantic content and its addresses [26], architecture requirements such as restrictions on the existence of services of third parties that will act as intermediaries with the purpose of being able to act if there are arguments (taking advantage of the SOAP multi-hop architecture).

### 3.2 Principles

The main principles that should be regarded in all the activities are: i) Incorporation of security concepts derived from the Security Risk Analysis and Management (attack, vulnerability, attacker, risk etc.) [9, 27]; ii) Iterative and incremental, so that the security of just a piece of functionality is analyzed in every iteration and just the WS security requirements of that piece are produced; iii) To encourage reusability of the security requirements and the analysis work performed within this stage through the creation, updating and support of a repository composed of, among others, WS security requirements templates, WS attack patterns grouped into profiles and specific WS security use cases [18, 21, 28].

### 3.3 Input

The development of intensive software systems starts with a description of the system operations and the highest level functional requirements [29, 30].

According to that, input is composed of a specification of the scope that we want to comprise during iteration (e.g.: if we have a definition of the use cases available, we can select those that we want to cover and use them as an input for iteration). If the current iteration is not the first one, we can consider those security risks that are of the highest priority and have not been solved in previous iterations yet, the business and security goals defined for the system as well as the part of the organizational security policy that we estimate that can influence the system design. These inputs can be obtained by applying methods such as QAWs (Quality Attribute Workshops) [30] or TOGAF (The Open Group Architecture Framework) [31].

### 3.4 Output

Adopted from SIREN [21] and following a series of basic principles for the definition of security requirements adapted for WS context based on [28], the output of this stage will be: i) System Security Requirements Specification (SyRS); ii) Security Software Requirements Specification (SRS); iii) System Tests Specification (SyTS); iii) Software Tests Specification (STS); iv) Interface Requirements Specification (IRS), extracted from SRS to avoid an extremely long document.

### 3.5 Actors

The actors taking part in this stage are the set of Stakeholders, RE (Requirements Engineering) Team and the Security Team. Stakeholders will act as sources of goals, policies, business high-level restrictions, etc. The RE Team will carry out the processes of elicitation, analysis, specification and validation of the security requirements together with the Security Team members.

### 3.6 Activities

**3.6.1 Elicitation.** The activity of elicitation will be supported by a detailed study of security for each WS business service identified and considered in the current iteration. Inspired by the risk analysis and management process known as Operationally Critical Attack, Asset, and Vulnerability Evaluation $^{SM}$ (OCTAVE) [28], we have defined the activity of elicitation according to the following steps:

1. Identification of the WS to be Protected, where the set of WS to be protected by the security requirements are to be identified (e.g.: consumer service, provider service, etc).

2. Attacker Profiles Identification. For each WS, we have to identify the potential types of attackers from the potential WS consumers. As a first step to identify potential attackers, we can invert every legitimate WS consumer into a potential attacker [32]. This is not enough; therefore, we must analyze possible attack scenarios coming from other type of attackers (e.g.: attackers of the underlying infrastructure used by the service to develop its functionality).

3. Potential Attacks Identification. Threats will take place in attack scenarios over WS (the interface with the service itself as well as the set of actions with internal elements such as databases or directories that the interface executes to complete that service) according to the type of SOA abstract interactions that the service must carry out (publishing, discover, binding and invocation). Normally, there will be two groups to be analyzed for each WS: the first one formed by binding and publishing processes of the WS and the second one formed by discovery, binding and invocation processes of the WS. Apart from defining the attack scenarios, we must define its associated security scenarios that specify how the WS should respond in order to prevent (or at best to mitigate) the attack. In Figure 3, QoS UML profile has been used to

model the level of quality (depicted with stereotype <<Asset>>) associated with certain business WS called ExampleHealthCarePatienInfoWS.

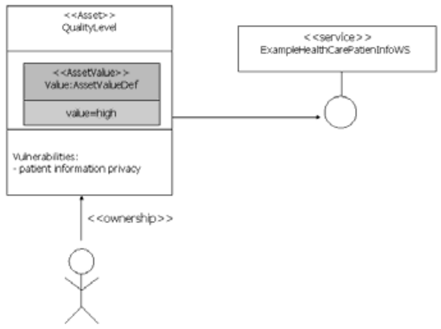


**Figure 3. Level of Quality modelled as an asset related to a WS by means of the QoS profile.**

In Figure 4, QoS UML profile modelling of unwanted incidents (exploited vulnerabilities by means of successful attacks) is shown

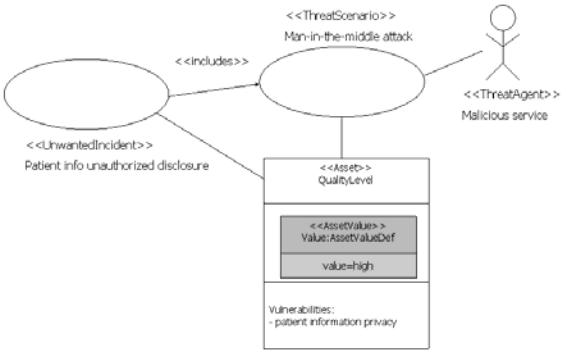


**Figure 4. The asset QualityLevel is related to the potential identified attack, which in turn is related to its attacker. In addition, the asset QualityLevel shows a relationship with the unwanted situation that may arise when its potential attack is successful**

4. Attack Impact Assessment. For each vulnerable WS, we must determine the negative impacts that could appear if the attacks against this WS would happen and how the impact of this attack could be spread to the services interacting with it as well as to the underlying infrastructure (e.g.: ERP systems, databases, etc.).

5. Security Risks Assessment and Priorization. Security risk is the potential risk of causing damage to an estimated WS from the sum (taking into account all the relevant threats) of the negative impact of the caused damage multiplied by the probability of this impact to happen [9]. In this way, we can find out which WS are more relevant in terms of security and we can dedicate more resources to them during iteration.

6. Select the Security Subfactors [33]. For each security risk of a given WS, we must select, with the objective of limiting the risk to an acceptable level, the set of relevant security subfactors (e.g.: authentication, authorization, etc.) that determine the types of security requirements necessary to reduce that risk to an acceptable level.

7. Security Requirement Specification that accomplishes the next steps:

a. Select from the repository of reusable requirements templates, the template or templates to be used for each security risk and associated subfactor. In our case, an example of template for information privacy could be as follows: *"The [WS consumer, WS provider, WS discovery] will guarantee the non revelation of [type | identifier] of information without the express consent of its owner to [WS consumer, WS provider, WS discovery] during the execution of [set of interaction/use cases] according to the criterion and measures specified in the table [table]"*.

b. Determine the security criterion in order to introduce its parameters into the template. The security criterion specific for the WS which the template will be used for, determines how the degree of presence of a certain security subfactor will be measured. The security criteria can be determined according to the services, types of attackers or identified attacks and security risks. In the template of our example, we have determined that the criterion will be: Minimum number of attributes kept private".

c. Determine the suitable security metrics that measures the existence of the chosen security criteria and to introduce the quality metrics into the template. In our case, metrics is a percentage.

d. According to the security risk identified for a certain service, to determine the minimum acceptable level of metrics for the chosen criterion, that limits to an acceptable level the associated risk and to introduce the template required level. In our example, the accepted level is 99.99 %.

e. Specify the security requirement by instancing the template from the selected parameters in the last three steps. An example of privacy requirement instance for an attribute service that takes part in a federation solution is shown below: *"The WS Provider of attributes of the HealthCare.example.com system, by means of the WS called ExampleHealthCarePatienInfoWS, will guarantee the non revelation of patients personal information to sophisticated attacks (as explained in TABLE 1) unless patients have given express consent that allows to reveal their personal data to the WS customer StatisticsGenerator of the system of XYZ company during the execution of the use cases Statistics and Foresight Generation"*.

**Table 1. Sample privacy requirement criterion, metrics and values.**

| | Minimum number of attributes kept private |
|---|---|
| StatisticsGenerator WS generates a report of the Patient Profile | 99.99% |
| | ... |
| StatisticsGenerator WS generates a report of occupational foresight | 99.99% |

8. Register the requirement together with the business goals or security policies which this requirement has been derived from and with the reasoning framework that explains its refining process (the group of artefacts used or produced such as threat trees, misuse cases, security use cases, etc.).

**3.6.2. Analysis.** The analysis activity basically consists of identifying the possible conflicts that could arise among the security requirements. The main steps are:

1. Identification of conflicts taking into account two possible perspectives: i) security requirements conflicts within composition scenarios [34]: If there are new services built by composition, we must verify that these new services do not violate any of the identified security requirements; ii) security requirements conflicts within integration scenarios [1, 35]: i) external services, governed by third parties, which we want to integrate with; ii) inherited systems that we want to offer an interface based on a WS for.

2. Elimination and refinement of redundant and ambiguous security requirements, respectively.

3. Security requirements Classification. The analysis classifies security requirements in terms of system security, software or interfaces specification and, also, it establishes traceability relationships between the different security requirements [21]. Moreover, we have added one more classification that groups security requirements depending on if they apply to a process of publishing, discovery, binding or invocation.

**3.6.3. Specification.** This activity basically consists of documenting the WS security requirements. Requirements specification, based on IEEE std. 1233, 12207.1, 830 standards, is supported by the idea of the use of a set of requirements documents and a structure of reusable documents defined in SIREN [21]. This strategy guarantees requirements reusability and implements traceability relationships, both inclusive and exclusive, among the security requirements. One of the most important aspects of specification is the development of the specification of software and system tests in a way that each system or software security requirement has an acceptance criterion defined from the moment of its definition.

**3.6.4. Validation & Verification.** The last stage defined in WSSecReq is security requirements validation and verification that will be developed in two main activities:

1. Internal verification that identifies the possible conflicts between security requirements and the rest of requirements and that detects ambiguous or poorly expressed specifications.

2. External validation: Due to the lack of alignment with the security policies that took part in the input to this process, we must verify that there is not a lack of alignment between security requirements and policies of identical nature. In addition, it will have to be carried out a validation of the security requirements together with the Stakeholders participants in this stage.

## 4. Related Work

At present, undoubtedly, the biggest effort is being carried out in the area of WS security standards and specification definition. This effort has caused the existence of a vast number of specifications and standards that make it difficult to handle and to know them by the organizations that would like to use them. The lack of a global vision has caused that many organizations have been very reticent to use this method since they have thought it was full of acronyms.

Concerning the definition of processes for WS secure system development, we can highlight the extension for the methodology oriented to Tropos agents and goals defined in [36]. Here, it is stated an adaptation of Tropos that lets us define the architecture that covers a certain set of requirements QoS of WS. EFSOC [37] is a event-driven framework driven for WS-based systems that defines a security model that can be easily fixed for systems in which the modifiability degree is high and therefore, they require a review and update of the authorization policies. In [6], a methodical and formal analysis based on "formal analysis of security-critical service-based software systems" is presented and in [7] a formal approach to the construction of service-based systems is presented.

None of these approaches proposes a method such as PWSSec that, from the business, system security goals and functional business and application WS patterns, can obtain a WS-based system. Moreover, none of these methods offers us facilities for the reusability of the generated products in a way that their practical applicability is guaranteed.

## 5. Conclusion and future research

This paper has presented the PWSSec process that facilitates the development and integration of security within WS-based systems. As far as authors know, there is not in the field of WS system research, a definition of a complete process comprising and taking into account all the stages of its life cycle.

Nowadays, we are applying PWSSec to two study cases carried out by two state-owned organizations. We hope that, as a result of this practical application, we could refine the stages of the process and enrich the products generated in it.

Some of the main aspects to be developed are the following: i) Complete the repository defined in WSSecReq stage with security requirements templates and specific attack patterns that comprise more WS security aspects; ii) Security requirements modelling and formal validation; iii) Develop evaluation areas and cost/benefit analysis of WSSecArch; iv) Complete the catalog of WSSecTech standards and specifications (nowadays, completed for authentication and perimeter security requirements); v) In the process of verification of policies compatibilities executed by WSSecKern, we still have to define whether two policies are semantically equivalent.

## 6. Acknowledgments

This research is part of the following projects: MESSENGER (PCC-03-003-1) financed by the "Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha" (Spain), CALIPO (TIC2003-07804-C05-03) and RETISTIC (TIC2002-12487-E) granted by the "Dirección General de Investigación del Ministerio de Ciencia y Tecnología" (Spain).

## 7. References

[1] C. Nott, *Patterns: Using Business Service Choreography In Conjuction With An Enterprise Service Bus*, 2004.

[2] IDC, "Cautious Web Services Software Adoption Continues; IDC Expects Spending to Reach $11 Billion by 2008," 2004.

[3] C. Gutiérrez, E. Fernández-Medina, and M. Piattini, "Web Services Security: is the problem solved?," *Information Systems Security*, vol. 13, pp. 22-31, 2004.

[4] M. Endrei, J. Ang, A. Arsanjani, S. Chua, P. Comte, P. Krogdahl, M. Luo, and T. Newling, *Patterns: Service-Oriented Architecture and Web Services*, 1st ed, 2004.

[5] M. Endrei, J. Ang, A. Arsanjani, S. Chua, P. Comte, P. Krogdahl, M. Luo, and T. Newling, *Patterns: Services Oriented Architectures and Web Services*, 2004.

[6] M. Deubler, J. Grünbauer, J. Jürjens, and G. Wimmel, "Sound Development of Secure Service-based Systems," presented at ICSOC'04, New York, USA, 2004.

[7] M. Deubler, J. Grünbauer, G. Popp, G. Wimmel, and C. Salzmann, "Towards a Model-Based and Incremental Development Process for Service-Based Systems," presented at International Conference on Software Engineering (IASTED SE 2004), Innsbruck, Austria, 2004.

[8] M. P. Papazoglou and D. Georgakopoulo, "Service-Oriented Computing," *Communications of the ACM*, vol. 46, pp. 25-28, 2003.

[9] D. G. Smith, "Common Concepts Underlying Safety, Security, and Survivability Engineering," SEI, Technical Note CMU/SEI-2003-TN-033, December 2003 2003.

[10] L. Bass and R. Kazman, "Architecture Based Development," Carnegie Mellon. Software Engineering Institute. CMU/SEI-99-TR-007, April 1999 April 1999.

[11] J. Jürjens, *Secure Systems Development with UML*: Springer, 2005.

[12] H. Yu, X. He, Y. Deng, and L. Mo, "Integrating Security Administration into Software Architecture Design," presented at International Conference on Software Engineering and Knowledge Engineering 2004, Banff, Canada, 2004.

[13] T. Grandison, M. Sloman, and I. College, "A Survey of Trust in Internet Applications," IEEE, Survey Fourth Quarter 2000 2000.

[14] B. Schneier, "Attack Trees: Modeling Security Threats," *Dr. Dobb's Journal*, 1999.

[15] WS-I, "Security Challenges, Threats and Countermeasures," 2005.

[16] G. Sindre and A. L. Opdahl, "Eliciting Security Requirements with Misuse Cases," presented at TOOLS-37'00, Sydney, Australia, 2000.

[17] I. Alexander, "Misuse Cases: Use Cases with Hostile Intent," *IEEE Computer Software*, vol. 20, pp. 58-66, 2003.

[18] A. P. Moore, R. J. Ellison, and R. C. Linger, "Attack Modelling for Information Security and Survivability," Software Engineering Institute 2001.

[19] OMG, "UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms," 2004.

[20] D. G. Firesmith, "Security Use Cases," *Journal of Object Technology*, vol. 2, pp. 53-64, 2003.

[21] A. Toval, J. Nicolás, B. Moros, and F. García, "Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach," *Requirements Engineering Journal*, vol. 6, pp. 205-219, 2001.

[22] I. C. Society, "Software Engineering Body of Knowledge," 2004.

[23] P. Krutchen, "The 4+1 View Model of Software Architecture," *IEEE Software*, pp. 42-50, 1995.

[24] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 2nd, ed: Addison-Wesley, 2003.

[25] H. Skogsrud, B. Benatallah, and F. Casati, "Model-Driven Trust Negotiation for Web Services," *IEEE Internet Computing*, pp. 45-51, 2003.

[26] E. Ferrari and B. Thuraisingham, "Security and Privacy for Web Databases and Services," E. 2004 and L. 2992, Eds. Berlin Heidelberg 2004: Springer-Verlag, 2004, pp. 17-28.

[27] C. J. Alberts, S. G. Behrens, R. D. Pethia, and W. R. Wilson, "OCTAVE Framework, Version 1.0," Carnegie Mellon. SEI. CMU/SEI-99-TR-017, September 1999 1999.

[28] D. G. Firesmith, "Engineering Security Requirements," *Journal of Object Technology*, vol. 2, pp. 53-68, 2003.

[29] L. Bass, F. Bachmann, R. J. Ellison, A. P. Moore, and M. Klein, "Security and Survivability Reasoning Frameworks and Architectural Design Tactics," SEI, Technical Note CMU/SEI-2004-TN-022, September 2004 2004.

[30] M. R. Barbacci, R. Ellison, A. J. Lattanze, J. A. Stafford, C. B. Weinstock, and W. G. Wood, "Quality Attribute Workshops (QWAs). Third Edition.," Carnegie Mellon. Software Engineering Institute. CMU/SEI-2003-TR-016, August 2003 2003.

[31] "TOGAF (The Open Group Architecture Framework). Versión 8.1. "Enterprise Edition"," The Open Group 2003.

[32] L. Liu, E. Yu, and J. Mylopoulus, "Security and Privacy Requirements Analysis within Social Setting," presented at 11th IEEE International Requirements Engineering Conference, Monterey Bay, CA, USA, 2003.

[33] D. G. Firesmith, "Specifying Reusable Security Requirements," *Journal of Object Technology*, vol. 3, pp. 61-75, 2004.

[34] K. M. Khan and J. Han, "A Process Framework for Characterising Security Properties of Component-Based Software Systems," presented at Australian Software Engineering Conference (ASWEC'04), 2004.

[35] G. Jonsdottir, L. Davis, and R. Gamble, "Designing Secure Integration Architectures," presented at ICCBSS 2003, 2003.

[36] M. Aiello and P. Giorgini, "Applying the Tropos Methodology for Analysing Web Services Requirements and Reasoning about Qualities of Services," *UPGRADE*, vol. 5, pp. 20-26, 2004.

[37] K. Leune and M. Papazaglou, "Specification and Querying of Security Constraints in the EFSOC Framework," presented at International Conference on Service Oriented Computing, New York City, USA, Willem-Jan van den Heuvel.