Conteúdo Conceitual e Aspectos Práticos da Ciência da Computação

Ernane Rosa Martins
(Organizador)

Atena Editora
Ano 2020

Conteúdo Conceitual e Aspectos Práticos da Ciência da Computação

Ernane Rosa Martins
(Organizador)

Atena Editora
Ano 2020

Editora Chefe
Profª Drª Antonella Carvalho de Oliveira
Assistentes Editoriais
Natalia Oliveira
Bruno Oliveira
Flávia Roberta Barão
Bibliotecária
Janaina Ramos
Projeto Gráfico e Diagramação
Natália Sandrini de Azevedo
Camila Alves de Cremo
Luiza Alves Batista
Maria Alice Pinheiro
Imagens da Capa
Shutterstock
Edição de Arte
Luiza Alves Batista
Revisão
Os Autores

Atena Editora
Ano 2020

Atena Editora
Ano 2020

Atena Editora
Ano 2020

Atena Editora
Ano 2020

Atena Editora
Ano 2020

Atena Editora
Ano 2020

Ciências Exatas e da Terra e Engenharias

Conteúdo conceitual e aspectos práticos da ciência da computação

Editora Chefe: Profª Drª Antonella Carvalho de Oliveira
Bibliotecária: Janaina Ramos
Diagramação: Camila Alves de Cremo
Correção: Vanessa Mottin de Oliveira Batista
Edição de Arte: Luiza Alves Batista
Revisão: Os Autores
Organizador: Ernane Rosa Martins

Dados Internacionais de Catalogação na Publicação (CIP)

C761 Conteúdo conceitual e aspectos práticos da ciência da computação / Organizador Ernane Rosa Martins. – Ponta Grossa - PR: Atena, 2020.

Formato: PDF
Requisitos de sistema: Adobe Acrobat Reader
Modo de acesso: World Wide Web
Inclui bibliografia
ISBN 978-65-5706-601-0
DOI 10.22533/at.ed.010201412

1. Computação. I. Martins, Ernane Rosa (Organizador). II. Título.

CDD 004

Elaborado por Bibliotecária Janaina Ramos – CRB-8/9166

Atena Editora
Ponta Grossa – Paraná – Brasil
Telefone: +55 (42) 3323-5493
www.atenaeditora.com.br
contato@atenaeditora.com.br

Atena Editora
Ano 2020

DECLARAÇÃO DOS AUTORES

Os autores desta obra: 1. Atestam não possuir qualquer interesse comercial que constitua um conflito de interesses em relação ao artigo científico publicado; 2. Declaram que participaram ativamente da construção dos respectivos manuscritos, preferencialmente na: a) Concepção do estudo, e/ou aquisição de dados, e/ou análise e interpretação de dados; b) Elaboração do artigo ou revisão com vistas a tornar o material intelectualmente relevante; c) Aprovação final do manuscrito para submissão.; 3. Certificam que os artigos científicos publicados estão completamente isentos de dados e/ou resultados fraudulentos.

Atena Editora
Ano 2020

APRESENTAÇÃO

A Ciência da Computação, traz inúmeros benefícios para a sociedade moderna, tais como: a criação de empregos, o desenvolvimento de novos equipamentos, o ganho de produtividade nas empresas e o acesso à informação. Os estudos realizados nesta área são aplicados em diversas outras áreas do conhecimento, proporcionando a resolução de diferentes problemas da sociedade, trazendo avanços significativos para a vida de inúmeras pessoas, fazendo com que cada vez mais estes profissionais sejam valorizados, requisitados e prestigiados no mercado de trabalho.

As empresas enxergam atualmente a necessidade cada vez maior de profissionais bem qualificados nesta área, a fim de que possam promover cada vez mais inovação, desenvolvimento e eficiência junto as empresas. Os estudos desta área focam no estudo de técnicas, metodologias e instrumentos computacionais, visando principalmente automatizar os processos e desenvolver soluções com o uso de processamento de dados. Desta forma, este livro, vem possibilitar conhecer os elementos principais desta ciência por meio do contato com alguns dos conceitos fundamentais desta área, apresentados por meio dos resultados relevantes alcançados nos trabalhos presentes nesta obra.

Dentro deste contexto, este livro aborda diversos assuntos importantes para os profissionais e estudantes desta área, tais como: a orientação dos alunos na busca e utilização de ferramentas computacionais e tipográficas de qualidade; aplicação de uma heurística baseada em Algoritmos Genéticos; uma análise qualitativa dos principais programas computacionais utilizados em fotogrametria computadorizada; os antipadrões de restrição de autorização em serviços Web orquestrados com BPEL4People; um sistema de atendimento automatizado, que inclui chat, chatbots e gerenciamento de atendentes; o sistema PSI, um prontuário online destinado a psicólogos; a Formação de Grupos de Alto Desempenho (FGAD) em Aprendizagem Colaborativa Baseada em Projetos (CPBL) usando Metodologias ágeis; a integração do método dos elementos finitos (Finite Element Method) - FEM associado a um Algoritmo Genético (GA) combinado com Lógica Nebulosa (Fuzzy) para o desenvolvimento de um filtro óptico destinado a sistemas DWDM (Dense Wavelength Division Multiplexing); o desenvolvimento de ferramenta de código aberto para uso em atividades de eletrônica durante o distanciamento social; um modelo de Algoritmo Genético para otimizar os parâmetros do COCOMO Básico; discussões sobre como e por que estudar automação hoje em dia; um processo de recomendação utilizando análise de sentimento sobre scripts de filmes e agrupando filmes de sentimentos similares; um modelo de previsão, com a utilização das

ferramentas de Redes Neurais Artificiais, para estimar o volume de uma usina hidrelétrica; o desenvolvimento de um Sistema de Informação Geográfica (SIG); um mapeamento sistemático da produção do conhecimento científico e tecnológico; a utilização de um jogo sério que pode auxiliar os profissionais de educação a identificar alunos com maior probabilidade de sofrerem de discalculia; e uma revisão da literatura quanto a utilização de aplicativos em síndromes coronarianas agudas.

Assim, os trabalhos apresentados nesta obra exemplificam a abrangência e importância da área de Ciência da Computação na atualidade, permitindo aos nossos leitores analisar e discutir os resultados encontrados. A cada autor, os mais sinceros agradecimentos, por contribuir com esta importante obra, e aos leitores, desejo uma excelente leitura, repleta de boas e relevantes reflexões.

Ernane Rosa Martins

SUMÁRIO

CAPÍTULO 5

BPEL4PEOPLE ANTI-PATTERNS: DISCOVERING AUTHORIZATION CONSTRAINT ANTI-PATTERNS IN WEB SERVICES

Data de aceite: 01/12/2020
Data de aceite: 21/08/2020

Henrique Jorge Amorim Holanda
Universidade do Estado do Rio Grande do Norte – UERN
Mossoró – RN
http://lattes.cnpq.br/2693553902815978

Carla Katarina de Monteiro Marques
Instituto Federal do Rio Grande do Norte – IFRN
Mossoró – RN
http://lattes.cnpq.br/8416423647851683

Francisca Aparecida Prado Pinto
Secretaria de Educação do Ceará – Seduc
Fortaleza – CE
http://lattes.cnpq.br/6762292340117608

Giovanni Cordeiro Barroso
Universidade Federal do Ceará – UFC
Fortaleza – CE
http://lattes.cnpq.br/1218783106447217

ABSTRACT. Despite the abundance of analysis techniques to discover anti- patterns in BPEL, there is hardly any support for authorization constraint errors in web services orchestrated by BPEL4People. Most techniques simply abstract from people (human user interactions), while people dependencies can be the source of all kinds of errors. This paper focuses on the discovery authorization constraint anti-patterns in web services orchestrated by BPEL4People. We present an analysis approach that is expressed in terms of rule card, the well- known, stable, adaptable, and effective model-checking techniques can be used to discover authorization constraint errors. Moreover, our approach enables a seamless integration of control-flow and authorization constraint verification.
KEYWORDS: Software Engineer, Software Quality, Antipattern, Web Service.

ANTIPADRÕES BPEL4PEOPLE: DESCOBRINDO RESTRIÇÕES DE AUTORIZAÇÃO EM SERVIÇOS WEB

RESUMO: Apesar da abundância de técnicas de análise para descobrir antipadrões em BPEL, existe uma carência no suporte para erros de restrição de autorização em serviços web orquestrados com BPEL4People. A maioria das técnicas simplesmente abstrai as interações humanas com o usuário, enquanto as dependências entre elas podem ser a fonte de muitos tipos de erros. Este artigo enfoca os antipadrões de restrição de autorização em serviços Web orquestrados com BPEL4People. Apresentamos uma abordagem de análise que faz a verificação de modelos bem conhecidos, estáveis, adaptáveis e eficazes, esta é usada para descobrir erros de restrição de autorização. Além disso, nossa abordagem permite uma integração perfeita de fluxo de controle.
PALAVRAS-CHAVE: Engenharia de Software, Qualidade de Software, Antipadrões e Serviço Web.

1 | INTRODUCTION

A BPEL4People is a software system that manages and executes operational processes involving people, applications, and/or information sources on the basis of process models. BPEL4People systems are driven by explicit process models, i.e., based on a process model, a system is configured that supports the modeled process where exist user interactions. In this paper, we primarily focus on the analysis of the models used to configure web services orchestrated by BPEL4People. However, our approach is also applicable to other software system that manages and executes operational processes involving people. In the last 15 years, many analysis techniques have been developed to analyze web services orchestrated by BPEL4People. Most analysis techniques focus on verification, i.e., the discovery of design errors. Although many process representations have been used or proposed, most researchers are using Petri Nets as a basic model [H.J.A Holanda and Serra 2010]. The flow-oriented nature of BPEL4People processes makes the Petri Net formalism a natural candidate for the modeling and analysis of work flows.

Unfortunately, lion's share of attention has been devoted to control-flow while ignoring other perspectives such as data-flow and resource allocation. Analysis techniques typically check for errors such as deadlocks, live locks, etc. while abstracting from data and uses. Existing approaches typically suffer from the following two problems:

1. they look at only one perspective in isolation (e.g., only control-flow), and
2. the types of errors they capture are usually not configurable and mainly driven by the verification algorithms themselves rather than by user requirements [Dumas et al. 2005].

To address some of the limitations of existing approaches, we propose a new analysis framework based on uses of "anti-patterns" expressed in terms of rule card. Assuming a rule card representation, we define anti-patterns related to the BPEL4People. The term "anti-patterns" was coined in 1995 by Andrew Koenig. He stated that "An anti-pattern is just like pattern, except that instead of solution it gives something that looks superficially like a solution but isn't one" [Moha et al. 2012]. The goal of anti-patterns is to formally describe repeated mistakes such that they can be recognized and repaired. In this paper, we use rule card to formalize our anti-patterns. This formalization can be used to dis- cover the occurrence of such anti-patterns in BPEL4People. Although not elaborated on in this paper, the same techniques can be used to define correctness notions related to the control-flow and check these in an integral way.

An example of an anti-pattern is Dangling Inputs Group of Users (DIGU). This anti-pattern describes the situation where some group of users' needs to be

attributed for one task, but either it has never been created. The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 introduces BPEL and BPEL4People. Section 4 introduces anti-patter. Section 5 presents the specification of authorization Con- strain anti-pattern in web services orchestrated with BPEL4People. Section 5 presents our proposed approach. Section 7 presents the experiments and results. Section 8 concludes the paper and gives suggestions of future works.

2 | RELATED WORK

In the past years, several catalogs to specify and detect anti-patterns of SOA [Moha et al. 2012], [Palma et al. 2014b], Rest [Palma et al. 2014a] and BPEL [Palma et al. 2013] services and languages have been proposed. In [Smith and Williams 2003] the authors detected and evaluated more new software antipatterns and in [Sinnig et al. 2005] the authors detected and evaluated patterns in model-based engineering. We can highlight the drawbacks of the current literature as follows:

- Anti-patterns and approaches to detect them were considered only for SOA [Palma et al. 2014b] , [Palma et al. 2014a] and [Palma et al. 2013] models; Extending BPEL Engines with BPEL4People [Holmes et al. 2008] [Martin Vasko 2007];
- A model checking approach to verify BPEL4WS workflows [Bianculli et al. 2007];
- Web Services Human Task (WS-HumanTask) [Ings et al. 2012];
- Access control and authorization constraints for WS-BPEL [Bertino et al. 2006]; Verifying BPEL workflows under authorization constraints [Zha 2006];
- Finally, there is no detection approach for detecting authorization constraint anti- patterns in web services orchestrated by BPEL4People anti-patterns until now; We focus on those issues with a solution to propose a concrete approach for specifying and detecting authorization constraint anti-patterns in web services orchestrated by BPEL4People.

3 | BPEL AND BPEL4PEOPLE

Business process management is designed to make business activity coordination easier and more cost effective [Bertino et al. 2006]. WS-BPEL and BPEL4People extension together coordinate the WS and human activities within business process. With the devel- opment of globalization organizations become

more dynamic and the underlying business process are frequently optimizing in today's business world. Adapting business processes to market changes and automating business services on demand are main necessities to facilitate business collaboration among existing and potential partners. In business process-oriented environment, a unified process specification language is significantly crucial in term of collaboration. WS-BPEL is one such language that provides the syntax and notations for specifying business processes behavior based on WS. Besides automatic WS, most business processes still require human interactions. BPEL4People addresses this important aspect to provide human actor support.

3.1 BPEL

WS have become widely accepted as the de-facto standard for distributed business applications [Holmes et al. 2008]. They bring maximum interoperability, use an open and exible architecture, and the implementation and complexity of a WS can be hidden towards a service requestor. Layered on top of these services, BPEL, the de-facto standard for orchestration, formally describes processes.

Web service composition refers to the creation of new (Web) services by combination of functionality provided by existing ones. This paradigm has gained significant attention in the WS community and is seen as a pillar for building service-oriented applications. A number of domain-specific languages for service composition have been proposed with consensus being formed around a process-oriented language known as WS-BPEL (or BPEL). The kernel of BPEL consists of simple communication primitives that may be combined using control-flow constructs expressing sequence, branching, parallelism, synchronization, etc. As a result, BPEL process definitions lend themselves to static flow-based analysis techniques. The business process execution language (BPEL) is an XML subset for specifying and executing business processes. As interactions are realized with WS for maximum interoperability between various heterogeneous systems, BPEL permits orchestration of WS.

3.2 BPEL4People

WS-BPEL focuses on business processes that orchestrate Web service interactions. How- ever, in general, business processes are comprised of a broad spectrum of activities that most often require the participation of people to perform tasks, review or approve steps and enter data — for example, a credit approval scenario that may require approval on certain transaction limits or activity levels. These human interactions are now addressed in the new specifications. Human user interactions are currently not covered by the Web Services Business Processes Execution Language (WS-BPEL), which is primarily de- signed to support automated

business processes based on WS. In practice, however, many business process scenarios require user interaction. So far, we've seen that user interaction in business processes can get quite complex. Although BPEL specification 1.1 (and the upcoming BPEL 2.0) doesn't specifically cover user interactions, BPEL is appropriate for human work flows. Work flow services that leverage the rich BPEL support for asynchronous services are created today. In this fashion, people and manual tasks become just another asynchronous service from the perspective of the orchestrating process and the BPEL processes stay 100% standard.

We now see the next generation of work flow specifications emerging around BPEL with the objective of standardizing the explicit inclusion of human tasks in BPEL processes. This proposal is called BPEL4People and was originally put forth by IBM and SAP in July 2005. Other companies, such as Oracle, have also indicated that they intend to participate in and support this effort.

The most important extensions introduced in BPEL4People are people activities and people links. People activity is a new BPEL activity used to define user interactions; in other words, tasks that a user has to perform. For each people activity, the BPEL server must create work items and distribute them to users eligible to execute them. People activities can have input and output variables and can specify deadlines. To specify the implementation of people activities, BPEL4People introduced tasks. Tasks specify actions that users must perform. Tasks can have descriptions, priorities, deadlines, and other properties. To represent tasks to users, we need a client application that provides a user interface and interacts with tasks: it can query available tasks, claim and revoke them, and complete or fail them.

To associate people activities and the related tasks with users or groups of users, BPEL4People introduced people links. People links are somewhat similar to partner links; they associate users with one or more people activities. People links are usually associated with generic human roles, such as process initiator, process stakeholders, owners, and administrators [Ings et al. 2012]. BPEL4People extends the capabilities of WS-BPEL to support a broad range of human interaction patterns, allowing for expanded modeling of business processes within the WS-BPEL language. BPEL4People is comprised of two specifications including:

- WS-BPEL Extension for People which layers features on top of WS--BPEL to de- scribe human tasks as activities that may be incorporated as first-class components in WS-BPEL process definitions.

- Web Services Human Task introduces the definition of stand-alone human tasks, including the properties, behavior and operations used to manipulate them. Capabilities provided by Web Services Human Task may be utilized by Web services-based applications beyond WS-BPEL processes.

3.2.1 Integrating Authorization Constraints

BPEL4People support features to exclude some users from performing a task because of some tasks they had done before or force some user to perform a sequence of tasks. We call such requirement as authorization constraint, as the term is widely used in access control literature. In this section we will use GSPN to express the authorization constraints to facilitate formal analysis. Two kinds of authorization constraints, namely “4-eyes principle” and “chained execution”, are proposed in BPEL4People specification. The “4-eyes principle”, also known as “separation of duty”, is a common scenario in many application areas when a decision must be made by two or more people independently of one another, often for the security reasons, and “chained execution” refers a process fragment where a sequence of steps must be executed by one person.

Separation of duty

The separation of duty (SoD) is a well-known principle in authorization to prevent fraud or error by requiring that at least two individuals are involved in some specific work. SoD is also useful when two persons have to co-operate in a work but none of them should know all the details. The basic form of SoD states that two given distinct tasks t1 and t2 must be performed by different individuals. This can be defined as states that person p0 cannot perform both t1 and t2. We can define variations of this similarly, e.g., “task t1 and t2 must be performed by different roles”. We can also define SoD constraint for a specific person, e.g., “person A cannot invoke both task t1 and t2”.

Binding of duty

“Binding of duty” (BoD) is the dual of SoD, which states that some distinct tasks must be performed by one person. BoD is used to define the responsibility of a person, e.g.: It states that if p0 performs t1, then p0 must also perform t2, and vice versa. SoD and BoD may be combined to define more complex constraints.

3.2.2 Overall Language Structure

This section explains the structure of BPEL4People extension elements, including the new activity type people activity, inline human tasks and people assignments.

The BPEL specification focuses on business processes the activities of which are assumed to be interactions with Web services, without any further prerequisite behavior. But the spectrum of activities that make up general purpose business processes is much broader. People often participate in the execution of business

processes introducing new aspects such as interaction between the process and user interface and taking into account human behavior. This specification introduces a set of elements which extend the standard BPEL elements and enable the modeling of human interactions, which may range from simple approvals to complex scenarios such as separation of duties, and interactions involving ad-hoc data.

The specification introduces the people activity as a new type of basic activity which enables the specification of human interaction in processes in a more direct way. The implementation of a people activity could be an inline task or a standalone human task defined in the WS-HumanTask specification. The syntax and state diagram of the people activity, and the coordination protocol that allows interacting with human tasks in a more integrated way is described. The specification also introduces X Path extension functions needed to access the process context. The goal of this specification is to enable portability and interoperability:

- Portability - The ability to take design-time artifacts created in one vendor's environment and use them in another vendor's environment.
- Interoperability - The capability for multiple components (process infrastructure, task infrastructures and task list clients) to interact using well--defined messages and protocols. This enables combining components from different vendors allowing seamless execution.

Out of scope of this specification is how processes with human interactions are deployed or monitored. Usually people assignment is accomplished by performing queries on a people directory which has a certain organizational model. The mechanism of how an implementation evaluates people assignments, as well as the structure of the data in the people directory is also out of scope.

Language Design

The BPEL4People extension is defined in a way that it is layered on top of BPEL so that its features can be composed with BPEL features whenever needed. All elements and attributes introduced in this extension are made available to both BPEL executable processes and abstract processes. This extension introduces a set of elements and attributes to cover different complex human interaction patterns, such as separation of duties, which are not defined as first-class elements.

Syntax

Informal syntax of a BPEL process and scope containing logical people groups, inline human tasks, and people activity. A BPEL4People process must use BPEL4People extension elements and elements from WS-HumanTask namespace. Therefore, elements from name spaces BPEL4People and WS-HumanTask must be

understood.

The element *<b4p:humanInteractions>* is optional and contains declarations of elements from WS-HumanTask namespace, that is *<htd:logicalPeopleGroups>*, *<htd:tasks>* and *<htd:notifications>.*The element *<htd:logicalPeopleGroup>* specifies a logical people group used in an inline human task or a people activity. The name attribute specifies the name of the logical people group. The name MUST be unique among the names of all logical people groups defined within the *<b4p:humanInteractions>* element.

The *<htd:task>* element is used to provide the definition of an inline human task. The syntax and semantics of the element are provided in the WS-HumanTask specification. The name attribute specifies the name of the task. The name MUST be unique among the names of all tasks defined within the *<htd:tasks>* element.

The *<htd:notification>* element is used to provide the definition of an inline notification. The syntax and semantics of the element are provided in the WS-HumanTask specification. The name attribute specifies the name of the notification. The name MUST be unique among the names of all notifications defined within the *<htd:notifications>* element.

The element *<b4p:peopleAssignments>* is used to assign people to process-related generic human roles.

Potential owners of a task are persons who receive the task so that they can claim and complete it. A potential owner becomes the actual owner of a task by explicitly claiming it. Before the task has been claimed, potential owners can influence the progress of the task, for example by changing the priority of the task, adding ad-hoc attachments or comments. All excluded owners are implicitly removed from the set of potential owners. A WS-HumanTask Definition MAY define assignment for this generic human role.

Excluded owners are are people who cannot become an actual or potential owner and thus they cannot reserve or start the task. A WS-HumanTask Definition MAY define assignment for this generic human role.

An actual owner of a task is the person actually performing the task. When task is performed, the actual owner can execute actions, such as revoking the claim, forwarding the task, suspending and resuming the task execution or changing the priority of the task. A WS-HumanTask Definition MUST NOT define assignment for this generic human role.

New activity type *<b4p:peopleActivity>* is used to model human interactions within BPEL processes. The new activity is included in the BPEL activity *<bpel:extensionActivity>* which is used as wrapper. The syntax and semantics of the

people activity are introduced in section 4 “People Activity”. BPEL scopes may also include elements from BPEL4People and WS-HumanTask name spaces except

for the *<b4p:peopleAssignments>* element.

All BPEL4People elements may use the element *<b4p:documentation>* to provide annotation for users. The content could be a plain text, HTML, and so on. The *<b4p:documentation>* element is optional and has the following syntax:

<b4p:documentation xml:lang=”xsd:language”>

...

< /b4p:documentation>

4 | ANTI-PATTERNS

Changes resulting from the evolution of orchestrated with BPEL4People can degrade its design of a web services and can often cause the appearance of poor solutions in the architecture: anti-patterns.

Patterns and anti-patterns exist to capture expertise, and to communicate knowledge. Anti-patterns are opposed to patterns which are good specifications (solutions) for recurring problems. An anti-pattern is a surface-level symptom that hints at the presence of a deeper, more serious problem. Anti-patterns could hinder [Palma et al. 2013] future maintenance and evolution of web services.

Antipatterns detection is therefore important to assess the design of web services and ease their maintenance and evolution. However, methods and techniques for detection authorization constraint anti-patterns in web services orchestrated by BPEL4People do not yet exist. Anti-patterns have many definitions: they are ”poor” solutions to recur- ring design problems, known solutions for solving problems, which are not practical and usable, and wrong practices that are quite commonly used. Antipatterns have symptoms and consequences and are induced by some root clauses.

The presence of anti-patterns must be identified in unsuccessful system and their absence shown in successful systems. Anti-patterns usually cause costly and complicated architectures, which are costly and difficult to maintain. Having anti-patterns in a system can hinder a project. Developers study anti-patterns to avoid pitfalls. But sometimes can create pitfalls in knowledge transfer if not applied appropriately. The idea that the use of anti-patterns in knowledge transfer may be a dangerous strategy if applied incorrectly than discovery and corrected before.

5 | SPECIFICATION OF AUTHORIZATION CONSTRAIN ANTI-PATTERN IN WEB SERVICES ORCHESTRATED WITH BPEL4PEOPLE

In this section we introduce the specification of 7 (seven) authorization constrain anti- pattern. These anti-patterns were adapted from the literature [Smith and Williams 2000], [Moha et al. 2012], [Palma et al. 2014b], [Palma et al. 2014a],

[Palma et al. 2013] and [Zha 2006].

- Dangling Inputs Group of Users (DIGU): DIGU is an anti-pattern where one group of users remains unused.
- Dangling Outputs Group of Users (DOGU): DOGU is an anti-pattern where one group of users is assigned for one service but it was not defined.
- Duplicated Group of Users (DGU): DGU corresponds to a group of highly similar users.
- Ambiguous Name of groups of users (ANGU): Ambiguous Name is an antipattern where the developers use the names of groups that are very short or long, include too general terms, or even show the improper use of verbs, etc. Ambiguous names are not semantically and syntactically sound and impact the discover ability and the re-usability of a Web service.
- Missing User (MU): When a user that performs a task is excluded to perform another task that he/she is not one user potential of this another task.
- Strongly Missing Groups of Users (SMGU): When who performs a task were the members of one group and another task excludes the members of this group, however no member of this group is a potential user of this another task.
- Weakly Missing Groups of Users (WMGU): When who performs a task was the members of one group and another task excludes the members of this group though some member of this group is a potential user of this another task and others not.

6 | APPROACH

With the aim of detecting anti-patterns in authorization constrain in web services orchestrated with BPEL4People, we used the approach shown in 1. This approach involves three major steps:

- Step 1 (Specification) concerns specifying rules for the detection of authorization constrain anti-patterns that, later on, will be applied on web services orchestrated with BPEL4People.
- Step 2 (Generation) transforms orchestrated with BPEL4People into an intermediary representation, i.e., more abstract and simplified, by filtering some process facts those are not required to apply rules to ease: (i) the implementation of the rules defined in the previous step and (ii) the further analysis of the processes.

- Step 3 (Detection) consists in applying the rules defined in Step 1 on the transformed processes in the previous step. Finally, a list of existing anti-patterns with the involved process fragments will be shown.

Static service antipatterns require only static analysis for their detection and, thus, only their structural properties are needed to detect them within the services. Coupled with the information about how these parts will be distributed across the web services orchestrated with BPEL4People, we use rules to detect the potential anti-patterns that may occur.

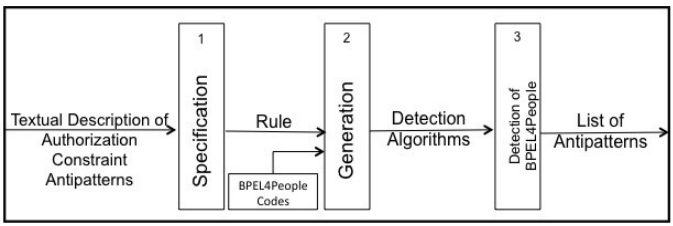


Figure 1. Proposed Detection Approach.

Our approach for detecting anti-patterns leverages static analysis to detecting antipatterns in authorization constrain in web services orchestrated with BPEL4People, Algorithm 1 presents one code BPEL4People for web services (WS-PurchSys) orchestrated.

7 | EXPERIMENTS AND RESULTS

This section applies the proposed detection approach for seven authorization constraint anti-patterns in Web services orchestrated by BPEL4People anti-patterns, specified in Sections 5.

- Step 1 concerns specifying rules for the detection of authorization constrain antipatterns that, later on, will be applied on Web services orchestrated with BPEL4People.

- Step 2 transforms orchestrated BPEL4People code into an intermediary representation, i.e., and more abstract and simplified code, by filtering some processes, those are not required to apply rules. This step consists also in implementing the rules showed in 2. We implement the rules in a modularized way, i.e., we implement each side of different logical operators (e.g., AND, OR) in a rule and join them afterwards to check the conformance with the defined conditions using the approach in [8].

- Finally, Step 3 applies the rules defined in Step 1 on the transformed processes in the previous step. The detection step follows the specifi-

cation of the rules and the transformation of BPEL4People codes into an intermediary representation, i.e., more abstract and simplified files XML. We show with a small scale experiment the effectiveness of our proposed approach.

We specify seven anti-patterns and we analyze and perform anti-pattern detection of these seven antipattern for three Web services orchestrated by BPEL4People using our approach.

- Employee-of-month: this Web services is started and as a first step, the people are determined that qualify as voters for the Employee of the month. Next, all the voters identified before get a chance to cast their votes. After that, the election result is determined by counting the votes casted. After the result is clear, two different people from the set of people entitled to approve the election either accepts or reject the voting result. In case any of the two rejects, then there is no Employee of the month elected in the given month, and the process ends. In case all approvals are obtained successfully, the employees are notified about the outcome of the election, and a to-do is created for the elected Employee of the month to prepare an inaugural speech. Once this is completed, the process completes successfully;

```
<process name="purchasing">
<b4p:humanInteractions><htd:tasks>
<htd:task name="manager_approve">
<htd:peopleAssignments><htd:potentialOwners>
<htd:user>Alan</htd:user></htd:potentialOwners>
</htd:peopleAssignments></htd:task>
<htd:task name="finance_approve">
<htd:peopleAssignments><htd:potentialOwners>
<htd:user>Ben</htd:user></htd:potentialOwners>
</htd:peopleAssignments></htd:task>
<htd:task name="purchase"><htd:peopleAssignments>
<htd:potentialOwners>
<htd:user>Ben</htd:user><htd:user>Cindy</htd:user><htd:user>Diana</htd:user>  </htd:potentialOwners><htd:excludedOwners>
<htd:getActualOwner("tns:finance_approve")
</htd:excludedOwners>
</htd:peopleAssignments></htd:task>
<htd:task name="notify_staff"><htd:peopleAssignments>
<htd:potentialOwners>
<htd:user>Diana</htd:user><htd:user>Edward</htd:user></htd:potentialOwners>
<htd:excludedOwners>
<htd:getActualOwner("tns:purchase")
</htd:excludedOwners>
</htd:peopleAssignments></htd:task>
</htd:tasks></b4p:humanInteractions>
<sequence> <receive createInstance="yes"/>
<extensionActivity>
<b4p:peopleActivity name="manager_approve"
outputVariable="result"/>
<b4p:localTask reference="tns:manager_approve"/>
</b4p:peopleActivity>
</extensionActivity>
<Switch>
<case>
<Invoque name “result"/>
<case>
<Invoque name “approved"/>
</Switch>
</sequence>
</process>
```

Algoritmo 1: Example of BPEL4People code for web services orches- trated.

- WS-PurchSys: this is a BPEL4People source code for a purchasing process, it is showed in Algorithm 1. Four tasks are defined: manager approve, finance approve, notify staff, and purchase. The potential owners of each task are: manager approve (Alan); finance approve (Ben); purchase (Ben, Cindy, Diana); notify staff (Diana, Edward). The excluded owner of purchase is the actual owner of finance approve. The excluded owner of notify staff is the actual owner of purchase; and
- Web-Service1: this Web services consists of two groups of users (”Group 1” and ”Group 2”) and five services (”Service 1”, ”Service 2”, ”Service 3” (”Service 3” excludes who performed the ”Service 2”), ”Service 4” (”Service 4” excludes who performed the ”Service 1”) and ”Service 5”).

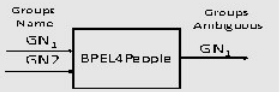
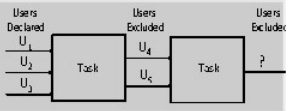
Anti-patterns	Rule	Diagrams
Dangling Inputs Group of Users (DIGU)	RULE_CARD DanglingInputsGroupUsers { RULE InputsGroupUsersUnused {DIGU ≥ 0} IF{DIGU≥ 0} THEN DanglingInputsGroupUsers}	
Dangling Outputs Group of Users (DOGU)	RULE_CARD DanglingOutputsGroupUsers { RULE OutputsGroupUsersUnused {DOGU ≥ 0} IF{DOGU ≥ 0} THEN DanglingOutputsGroupUsers}	
Dangling Group of Users (DGU)	RULE_CARD DanglingGroupUsers { RULE GroupUsersUnused {DGU ≥ 0} IF{DGU≥ 0} THEN Dangling GroupUsers }	
Ambiguous Name of groups of users (ANGU)	RULE_CARD AmbiguousNameGroupsUsers { RULE AmbiguousName {ANGU ≥ 0} IF{ANGU ≥ 0} THEN AmbiguousNameGroupsUsers}	
Missing User (MU)	RULE_CARD DanglingOutputResource { RULE OutputResource Unused {MU ≥ 0} IF{MU ≥ 0} THEN DanglingOutputResource}	
Strongly Missing Groups of Users (SMGU)	RULE_CARD StronglyMissingGroupsUsers { RULE StronglyMissing {SMGU ≥ 0} IF{SMGU ≥ 0} THEN StronglyMissingGroupsUsers}	
Weakly Missing Groups of Users (WMGU)	RULE_CARD WeaklyMissingGroupsofUsers { RULE WeaklyMissing {WMGU ≥ 0} IF{WMGU ≥ 0} THEN WeaklyMissingGroupsofUsers}	

Figure 2. Rules specifications and diagrams.

We could detect the seven authorization constrain antipatterns specified in Section 5 in the three Web services orchestrated with BPEL4People given as an examples.

We present in Table 1 the anti-patterns detection results for the three Web

services orchestrated with BPEL4People followed by some discussion about accuracy and performance detection. Initially some anti-patterns were not detected in these Web services orchestrated with BPEL4People. Subsequently some changes were inserted in these codes to test the accuracy and performance of the detection algorithm then anti-patterns were detected.

	Chessboard top view	Chessboard perspective view
Selection with side movements	6.02 ± 5.22	7.01±6.84
Selection with in-depth movements	6.29±4.99	12.22±11.33
Manipulation with side movements	4.66± 4.94	3.47±2.20
Manipulation with in-depth movements	5.71 ±4.55	5.37 ±3.28

Table 1. Experiments results.

Through our experiment, we aim to show the accuracy and performance of the detection algorithms in terms of precision, recall and detection time, presented in Table 1. Antipattern detection algorithms have at least a precision of 99%. Assuming that the anti- patterns have a negative impact on the design, we target a recall of 100% for anti-patterns, which ensures that we do not miss any existing anti-patterns. The precision concerns the detection accuracy of our specified rules and the corresponding detection algorithms.

The primary threat to the validity concerns the external validity of our results, i.e., generalizing the proposed approach to other Web services orchestrated with BPEL4People. We perform specification and detection of seven authorization constrain anti-patterns. However, we ran the experiment on a set of three Web services orchestrated with BPEL4People to minimize this threat. We plan to replicate the experiment on others Web services orchestrated with BPEL4People as a future work. We do not execute the Web services orchestrated with BPEL4People, hence analyze the processes statically, and the injections were performed internally, which are threats to the internal validity. However, we plan to execute the Web services orchestrated with BPEL4People flows to dynamically analyze them in the future.

The subjective nature to define rule cards is a threat to construct validity. However, we lessen this threat by defining the rule cards after a thorough literature review. Finally, the conclusion validity threats refer to the relation between the

treatment and the outcome. We paid full attention not to violate the assumptions of the performed statistical tests. We mainly used non-parametric tests that do not require making any presumption about the data distribution. The threats to internal validity concern the possibility of replicating this study. To minimize this threat, we provide all the details required to replicate the study, including the source code repositories and the raw data used to compute the statistics on our Webservices. One major challenge to minimize the threat to the external validity is the very limited availability of open-source Web services orchestrated with BPEL4People.

Finally, a list of anti-patterns, among the seven specified, existing in three Web services orchestrated by BPEL4People is showed in Table 1.

8 | CONCLUSIONS AND FUTURE WORK

In this paper we assessed the authorization constrain in Web services orchestrated with BPEL4People and eased the maintenance and evolution of these architectures.

We proposed an approach to specify authorization constrain anti-patterns and detect them in Web services orchestrated with BPEL4People. We present our static analysis-based approach to detect antipatterns in BPEL4People codes. We have also proposed to assess the design and QoS of Web services orchestrated with BPEL4People. To achieve this goal, we propose an approach to specify authorization constrain anti-patterns and detect them in Web services orchestrated with BPEL4People. Finally, we want to quantify the impact of detection anti-patterns on the maintenance and evolution of Web services orchestrated with BPEL4People.

The contribution of this paper is specifying and detect authorization constrain antipatterns in Web services orchestrated with BPEL4People, by leveraging static code analysis and information about prospective deployment of the application components using Web services orchestrated with BPEL4People. We applied and validated the detection algorithms using three Web services orchestrated by BPEL4People showed in Section 7, in terms of precision and recall. We specified and detected seven authorizations constrain antipatterns in an initial experiment with tree Web services orchestrated by BPEL4People. Results showed that this approach has an average detection precision of more than 97% and recall of 100%. Hence, we make a strong recommendation that designers be exposed to antipatterns repeatedly until the pattern is well established. Only then should anti-patterns be used to strengthen the pattern in the designer's mind.

ACKNOWLEDGEMENTS

We would like to thank the CNPQ and Canada Chair.

REFERENCES

(2006). *Verifying BPEL Workflows Under Authorisation Constraints*, volume 4102 of *Lecture Notes in Computer Science*. Springer.

Bertino, E., Crampton, J., and Paci, F. (2006). Access control and authorization con- straints for ws-bpel. In *Web Services, 2006. ICWS '06. International Conference on*, pages 275–284.

Bianculli, D., Ghezzi, C., and Spoletini, P. (2007). A model checking approach to verify bpel4ws workflows. In *Service-Oriented Computing and Applications, 2007. SOCA '07. IEEE International Conference on*, pages 13–20.

Dumas, M., van der Aalst, W. M., and ter Hofstede, A. H. (2005). *Process-aware In- formation Systems: Bridging People and Software Through Process Technology*. John Wiley & Sons, Inc., New York, NY, USA.

H.J.A Holanda, J. Merseguer, G. C. and Serra, A. B. (2010). Performance evaluation of web services orchestrated with ws-bpel4people. In *International Journal of Computer Networks & Communications*, volume 2, pages 117–134. AIRCC Publishing Corporation.

Holmes, T., Vasko, M., and Dustdar, S. (2008). Viebop: Extending bpel engines with bpel4people. In *PDP*, pages 547–555. IEEE Computer Society.

Ings, D., Cle´ment, L., Ko¨nig, D., Mehta, V., Mueller, R., Rangaswamy, R., Rowley, M., and Trickovic, I. (2012). Web services human task (ws-humantask) specification version 1.1. OASIS Committee Specification Draft 12 / Public Review Draft 05.

Moha, N., Palma, F., Nayrolles, M., Conseil, B., Gue´he´neuc, Y.-G., Baudry, B., and Je´ze´quel, J.-M. (2012). Specification and Detection of SOA Antipatterns. In Liu, C., Ludwig, H., Toumani, F., and Yu, Q., editors, *Service-Oriented Computing*, volume 7636 of *Lecture Notes in Computer Science*, pages 1–16. Springer Berlin Heidelberg.

Palma, F., Dubois, J., Moha, N., and Gue´he´neuc, Y. (2014a). Detection of REST patterns and antipatterns: A heuristics-based approach. In *Service-Oriented Computing - 12th International Conference, ICSOC 2014, Paris, France, November 3-6, 2014. Proceed- ings*, volume 8831 of *Lecture Notes in Computer Science*, pages 230–244. Springer.

Palma, F., Moha, N., and Gueheneuc, Y.-G. (2013). Detection of process antipatterns: A bpel perspective. In *Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2013 17th IEEE International*, pages 173–177.

Palma, F., Moha, N., Tremblay, G., and Gue´he´neuc, Y. (2014b). Specification and de- tection of SOA antipatterns in web services. In *Software Architecture - 8th European Conference, ECSA 2014, Vienna, Austria, August 25-29, 2014. Proceedings*, pages 58–73.

Sinnig, D., Gaffar, A., Reichart, D., Forbrig, P., and Seffah, A. (2005). Patterns in model-based engineering. In Jacob, R., Limbourg, Q., and Vanderdonckt, J., editors, *Computer-Aided Design of User Interfaces IV*, pages 197–210. Springer Netherlands.

Smith, C. U. and Williams, L. G. (2000). Software performance antipatterns. In *Proceed- ings of the 2Nd International Workshop on Software and Performance*, WOSP '00, pages 127–136, New York, NY, USA. ACM.

Smith, C. U. and Williams, L. G. (2003). More new software antipatterns: Even more ways to shoot yourself in the foot. In *29th International Computer Measurement Group Conference, December 7-12, 2003, Dallas, Texas, USA, Proceedings*, pages 717–725. Computer Measurement Group.

ÍNDICE REMISSIVO

O

P

Q

R

S

T

V

W

Conteúdo Conceitual e Aspectos Práticos da Ciência da Computação

www.atenaeditora.com.br
contato@atenaeditora.com.br
@atenaeditora
www.facebook.com/atenaeditora.com.br

Atena Editora
Ano 2020

Conteúdo Conceitual e Aspectos Práticos da Ciência da Computação

www.atenaeditora.com.br
contato@atenaeditora.com.br
@atenaeditora
www.facebook.com/atenaeditora.com.br

Atena Editora
Ano 2020