

**Lidia Artamonova**  
**Tatyana Evtushenko**  
**Tatyana Shilova**

# **Academic Writing for IT Students**

Томск  
Издательство Томского государственного университета  
2023

УДК 811.11+004.9(075.8)  
ББК 81.2Англ-923

**Artamonova L., Evtushenko T., Shilova T.**

**Academic Writing for IT Students. – Томск: Издательство  
Томского государственного университета, 2023. – 141 с.**

ISBN 978-5-907572-76-8

This textbook is intended for Master and PhD Information Technology students (B1-C1 level of English proficiency). The instructions of how to write a research paper in English and the relevant exercises are given. The peculiarities of each section of a paper are presented. The exercises are based on real science materials taken from peer-reviewed journals. The subject area covers a wide scope of different Information Technology domains.

**УДК 811.11+004.9(075.8)  
ББК 81.2Англ-923**

ISBN 978-5-907572-76-8

© Artamonova L., Evtushenko T.,  
Shilova T., 2023  
© Томский государственный  
университет, 2023

# CONTENTS

<b>PREFACE .....</b>	<b>5</b>
 <b>Part I. WRITING AN INFORMATION TECHNOLOGY RESEARCH PAPER.....</b>	 <b>7</b>
<b>1.1. TITLE AND KEYWORDS .....</b>	<b>7</b>
<b>1.2. ABSTRACT .....</b>	<b>11</b>
<b>1.3. THE STRUCTURE OF THE INTRODUCTION .....</b>	<b>32</b>
<b>1.4. THE STRUCTURE OF THE MAIN BODY OF THE PAPER.....</b>	<b>66</b>
<b>1.5. THE STRUCTURE OF THE CONCLUSIONS.....</b>	<b>69</b>
<b>1.6. ACKNOWLEDGEMENTS.....</b>	<b>85</b>
 <b>Part II. SUBMITTING A PAPER TO A CONFERENCE.....</b>	 <b>88</b>
<b>2.1. CONFERENCE TITLES .....</b>	<b>88</b>
<b>2.2. CALL FOR PAPERS .....</b>	<b>96</b>
<b>2.3. ABSTRACT AND PAPER SUBMISSION.....</b>	<b>105</b>
<b>APPENDIX 1: Latin Abbreviations in Academic Writing.....</b>	<b>112</b>
<b>APPENDIX 2: The Table of Irregular Verbs .....</b>	<b>114</b>
<b>APPENDIX 2: Passive voice .....</b>	<b>118</b>
<b>APPENDIX 3: Non-finite forms of the verb .....</b>	<b>119</b>
<b>APPENDIX 4: Word Formation. Affixation .....</b>	<b>120</b>
<b>APPENDIX 5: Linking Words and Phrases .....</b>	<b>126</b>

<b>MATERIALS USED FOR ANALYSIS.....</b>	<b>128</b>
<b>REFERENCES.....</b>	<b>130</b>
<b>Internet resources.....</b>	<b>130</b>
<b>Keys to the exercises .....</b>	<b>131</b>
<b>Introduction .....</b>	<b>132</b>
<b>Conclusions .....</b>	<b>139</b>

## **PREFACE**

The textbook aims to develop postgraduate students' writing skills necessary for science communication helping them to better understand authentic texts in different domains of information technology and report their research findings.

An information technology research paper written for publication in one of International scientific journals, like any other research paper, is supposed to consist of the following sections:

- Title;
- Abstract;
- Introduction;
- Main body;
- Conclusions;
- Acknowledgements;
- References.

However, the number of sections and their names can vary in different journals.

The paper written in English must be well presented at a conference when needed. Thus, some information and advice consistent with appropriate self-study activities are provided for students who are planning their scientific career.

The textbook is structured as follows.

- The Preface gives a review of issues that must be studied when building skills in academic writing in English.

- Part I (Sections 1.1-1.6) presents the structure of a research paper, the knowledge of which is essential for young researchers. Every section gives a full explanation of how to write each part of the scientific paper including both methodological recommendations and

a sufficient number of exercises that can be chosen by a teacher depending on students' language skills, proficiency and syllabus.

- Part II (Sections 2.1-2.3) provides authentic materials from conference announcements. The analysis of a number of Calls for papers boosts students' vocabulary, which helps them to easily find the right conference, register and submit a paper in accordance with their scientific interests.

## **Part I. WRITING AN INFORMATION TECHNOLOGY RESEARCH PAPER**

### **1.1. TITLE AND KEYWORDS**

#### ***Focus on theory***

Every paper starts with the Title, after which one can find the names of the authors, their affiliation, post and e-mail addresses and the institution where research was done. Then goes the Abstract, at the end of which there is a list of keywords.

Every word in the title should be meaningful. The title is supposed to be simple and accurately reflect the investigation in order to attract the right kind of readers. As a matter of fact, it should be short and indicate what is written in the paper itself.

Usually titles are incomplete sentences but they can be in a question form as well. If the title is an incomplete sentence, no period is placed at the end of the sentence. If the title is a question, which is rather rare, there must be a question mark at the end of it.

All the words in the title are often capitalized, apart from articles, prepositions and conjunctions. This is always specified in a template. If the title consists of two parts, they are divided by a colon. Remember that the first word after the colon is written with a capital letter no matter what part of speech it may be.

One important issue is the use of the articles in the title. The title is more often an incomplete sentence (containing nouns, infinitives, gerunds), that is why it should conform to the rules of English Grammar. Consequently, you can use either definite or indefinite articles if you need.

The prepositions that are frequently used in the title are *by* (how something is done), *for* (for the purpose of), *from* (the origin of),

*in/on* (where something is located; what something regards), and *of* (belonging to, regarding).

**Note:** The title should consist of 6-10 words and include the words relevant for your research area and attract the reader's attention.

The keywords are more often nouns. Remember to choose nouns very carefully. Avoid using a string of nouns otherwise clarity might be lost. The adjectives that have been chosen should show the uniqueness of the work.

**Exercise 1.1.** Read the title and keywords below and answer the following questions.

- How many words does the title consist of?
- What parts of speech does it consist of?
- Are there any keywords in the title?
- What parts of speech are usually capitalized?
- Are there any prepositions? What do they mean?

DOUBLY STOCHASTIC POISSON PROCESS  
AND THE PRICING OF CATASTROPHE REINSURANCE  
CONTRACT

Ji-Wook Jang

Actuarial Studies, The University of New South Wales  
Sydney NSW **2052**, Australia

Tel.: +61 2 9385 3360, Fax: +61 2 9385 1883, Email:

j.jang@unsw.edu.au

**KEYWORDS**

Doubly stochastic Poisson process. Shot noise process. Piecewise deterministic **Markov** process theory. Stop loss reinsurance contract. Equivalent martingale probability measure. Escher transform.



**Exercise 1.2.** Read the title and keywords below and answer the questions.

- Does the title attract the right kind of readers? Why?
- What parts of speech does it consist of?
- Does it have a definite and concise indication of what is written in the paper itself?
- What information about the authors is given? Is it complete or not?

A Cross-Cultural Comparative Study of Users' Perceptions  
of a Webpage: With a Focus on the Cognitive Styles of  
Chinese, Koreans and Americans  
Ying Dong \* and Kun-Pyo Lee  
Industrial Design Department, KAIST, Daejeon, Korea

Keywords – Cross-Cultural Study, Cognitive Style, Webpage  
Perception, Eye Tracking.

**Exercise 1.3.** Read the information below and answer the questions.

- What keywords for the paper with this title might be?
- What parts of speech are capitalized?
- Are there punctuation marks in the title?

Design pattern recovery through visual language parsing and source  
code analysis

Andrea De Lucia, Vincenzo Deufemia, Carmine Gravino \*, Michele  
Risi

Dipartimento di Matematica e Informatica, Università di Salerno,  
Via Ponte Don Melillo, 84084 Fisciano (SA), Italy

a r t i c l e i n f o

Article history:

Received 23 October 2008

Received in revised form 5 February 2009

Accepted 5 February 2009

Available online 20 February 2009

*Self-study activity*

Watch the video How to Write a Research Paper Title

<https://www.youtube.com/watch?v=F11q-I3bE0c>

Answer the questions:

Why is it important to create the perfect title for the research paper?

Is it the first piece of information that the editor sees?

How many words does the title consist of?

What parts of speech does it consist of?

Are there any keywords in the title?

What parts of speech are usually capitalized?

Write the draft of the Title and the list of keywords for the research paper of your own.

## 1.2. ABSTRACT

### *Focus on theory*

A. Wallwork distinguishes four main types of abstracts (Wallwork, 2013). Their function is to summarize the research. That is why they are sometimes called Summaries.

An *unstructured* abstract consists of one paragraph (100-250 words) and contains a very brief summary of the paper.

A *structured* abstract consists of two or even more paragraphs and also contains a brief summary of the paper.

An *extended* abstract is organized as a full paper (e.g. Introduction, Methods, Discussion...), but much shorter (two to four pages). According to the requirements the abstract is optional.

A *conference* abstract is usually an extended abstract (up to 500 words) written for the conference.

The type of abstract depends on the journal or conference where you are going to submit your paper. The instructions to the authors are always given in a template.

It is common practice to write the draft of the abstract. It can be used as a plan of the paper. When the paper has been completed, the abstract is written.

In the abstract, the author should describe:

- **aim of research;**
- **methods of study;**
- **results of research;**
- **conclusions and/or recommendations.**

The styles that are used for writing abstracts can be personal or impersonal.

Personal way:

*We discovered that...*  
*The authors discovered that...*

Impersonal way:

*It was discovered that ...*

The most commonly used tenses in abstracts are:

The p r e s e n t s i m p l e (*we show*)

The p a s t simple (*we showed*)

The present perfect (*we have shown*)

The link words are very important in the abstract. The most frequently used are *however, otherwise, instead, moreover, also, in this paper, consequently in addition.*

### ***The use of keywords***

The keywords are of great importance because the readers stick to the keywords in their search for information. That is why the authors must have the keywords not only in the title but in the abstract of the article as well. It is recommended that the keywords should not be repeated more than 3 times in the abstract.

## **PART A**

***Exercise 2.1.*** Read and analyze the Abstract below.

### **ABSTRACT**

Distributed object technology *makes* object oriented programming even more powerful and efficient by making use of the objects that are available on different systems connected on a heterogeneous network, in addition to the locally defined objects. Wide range of hardware platforms and variety of operating systems can be inter-connected at the software level and deliver a more robust and comprehensive solution for today's Internet driven businesses. Internet itself plays an important role as a backbone for this technology. As the scope of the application code is not restricted to single source, platform or language, maximum attention must be paid

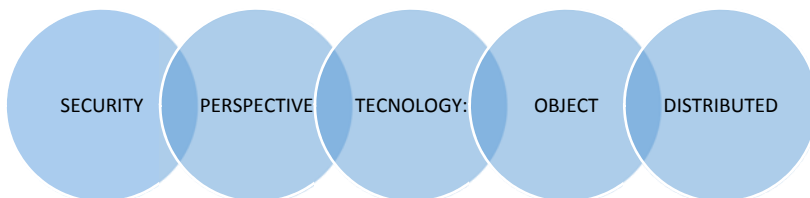
to the security of the applications. The objective of this paper is to give a brief introduction to distributed object technology and an overview of security features available in Microsoft.NET and CORBA. The paper explains the architecture of .NET and covers some of its key security concepts like Security Policy, Code Access Security, Role Based Security, Verification and Stackwalk. It also explains CORBA and its security concepts like CORBA Security Services, Security Specifications, Security Policy, Domain Access Policy and Delegation. The paper concludes by explaining the way in which some key security concerns are addressed in .NET and CORBA.

**Exercise 2.2.** Answer the questions.

- What is the aim of the research?
- Are there any methods of study given?
- What are the results of research?
- Is the issue of the research still relevant?

**Exercise 2.3.** Make the list of keywords for the Abstract given above.

**Exercise 2.4.** Make the heading of the paper using the words given below:



**Exercise 2.5.** Think of the definitions for the highlighted words in the Abstract.

**Exercise 2.6.** Fill in the prepositions *at of on to (2) in* to make word combinations complete.

- available ... different systems
- inter-connected ... the software level...addition ...the locally defined objects
- making use ... the objects
- a brief introduction ...distributed object technology

**Exercise 2.7.** Match the verbs with the nouns.

1.to give	A. a problem, an area, an issue, the schedule, concerns
2.to make	B. profits, letters, technology, books
3.to distribute	C. a chance, a lecture, an idea, an opinion, priority, evidence, notice, rise, a go, thought, an introduction
4. to cover	D. costs, share, losses, a concept, the issue, the topic
5. to address	E. an appointment, efforts, a decision, a choice, an exception, progress, a profit a suggestion, an attempt, a point, changes, use

**Exercise 2.8.** Tell your partner/group what kind of things you could.

distribute	address	make	cover	give
------------	---------	------	-------	------

**Grammar focus: *as***

<b>Comparison</b>	<b>As</b>	<b>a noun</b>
<b>Cause</b>		<b>a clause</b>

**Exercise 2.9.** What does as mean in these sentences?

1. Internet itself plays an important role *as* a backbone for this technology.

2. As the scope of the application code is not restricted to single source, platform or language, maximum attention must be paid to the security of the applications.

3. A design pattern can be seen as a set of classes, related through inheritances, aggregations and delegations,

4. According to (Gamma et al., 1995), design patterns are classified as structural ones.

5. This technique is known as invisible digital watermarking as human eye can not identify the change in the envelope image and the enveloped (Produced after LSB replacement) image.

**Exercise 2.10.** Boost your vocabulary with the words from the Abstract.

Noun	Verb	Adjective	Adverb
Technology	concern	heterogeneous	locally
programming	distribute	powerful	
variety	define	efficient	
backbone	restrict	available	
scope		robust	
objective		comprehensive	
overview			
Feature			
Specification			
delegation			

## PART B

**Exercise 2.11.** Read and analyze the Abstract below.

### **Visual Cryptography Scheme for Color Image Using Random Number with Enveloping by Digital Watermarking**

Shyamalendu Kandar Indian Institute of Engineering Science and Technology, Shibpur , Arnab Maiti, Bibhas Chandra Dhara Jadavpur University

#### **Abstract**

Visual Cryptography is a special type of encryption technique to obscure image-based secret information which can be decrypted by Human Visual System (HVS). This cryptographic system encrypts the secret image by dividing it into  $n$  number of shares and decryption is done by superimposing a certain number of shares( $k$ ) or more. Simple visual cryptography is insecure because of the decryption process done by human visual system. The secret information can be retrieved by anyone if the person gets at least  $k$  number of shares. Watermarking is a technique to put a signature of the owner within the creation.

In this current work we have proposed Visual Cryptographic Scheme for color images where the divided shares are enveloped in other images using invisible digital watermarking. The shares are generated using Random Number.

**Exercise 2.12.** Make up questions starting with **What (2) How (2) Why.**

**Exercise 2.13.** Fill in the appropriate link words (you will not need all of them): *firstly, secondly, finally, in addition to, also, moreover, otherwise, while, but, consequently, in this paper, herein.*



**Exercise 2.14.** Find the derivatives of the words given below in the text and translate them.

- Encrypt \_\_\_\_\_
- Inform \_\_\_\_\_
- Impose \_\_\_\_\_
- Own \_\_\_\_\_
- Create \_\_\_\_\_
- Sign \_\_\_\_\_

**Exercise 2.15.** Match the terms with their definitions.

1. A technique	A. the method of embedding data into digital multimedia content.
2. Visual cryptography	B. a part of any visual content
3. A random number	C. a cryptographic technique which allows visual information (pictures, text, etc.) to be encrypted in such a way that the decrypted information appears as a visual image.
4. Share	D. a particular method of doing an activity, usually a method that involves practical skills.
5. Digital watermarking	E. a number generated using a large set of numbers and a mathematical algorithm which gives equal probability to all numbers occurring in the specified distribution

### Grammar focus: Gerund 1.as a noun 2.after prepositions

1. The –ing form of the verb can be used as a noun.
2. The - ing form is used after prepositions. This includes *to* when it is a preposition and not part of the Infinitive

**Exercise 2.16.** What do you think the highlighted words mean?

1. This cryptographic system encrypts the secret image by dividing it into n number of shares and decryption is done by superimposing a certain number of shares(k) or more.
2. Watermarking is a technique to put a signature of the owner within the creation.
3. In this paper we have presented/present an approach for automating the testing process for Android applications.
4. Object-Role Modeling (ORM) is a method for modeling and querying an information system at the conceptual level, and mapping between conceptual and logical (e.g. relational) levels.
5. The detailed picture provided by ORM is desirable in developing and transforming a model.
6. To add more security to this scheme we have proposed a technique called digital enveloping.
7. One advantage of using Probe kit is that a probe collects data about the portion of code we are interested in.
8. Section 8 discussed the possible threats to the validity of the results before concluding the findings in Section 9.
9. The exceptions were mostly about querying on multiple non-existent tables in the app's SQLite database.
10. Thus, app developers are unable to reproduce the crash during the exploration, to conduct a regression test after fixing the bug, or to execute the same test under different environments.

## PART C

**Exercise 2.17.** Read and analyze the Abstract below.

### Abstract

Users increasingly 1. *have relied/rely* on mobile applications for computational

needs. Google Android is a popular mobile platform, hence the reliability of Android applications 2. *became/is becoming* increasingly important. Many Android correctness issues, however, fall outside the scope of traditional verification techniques, as they are due to the novelty of the platform and its GUI-oriented application construction paradigm. In this paper we 3. *have presented/present* an approach for automating the testing process for Android applications, with a focus on GUI bugs. We first 4. *conducted / conduct* a bug mining study to understand the nature and frequency of bugs affecting Android applications; our study 5. *finds /has found* that GUI bugs are quite numerous. Next, we 6. *present/ have presented* techniques for detecting GUI bugs by automatic generation of test cases, feeding the application random events, instrumenting the VM, producing log/trace files and analyzing them post-run. We 7. *are show/ show* how these techniques helped to re-discover existing bugs and find new bugs, and how they could be used to prevent certain bug categories. We 8. *have believed /believe* our study and techniques have the potential to help developers increase the quality of Android applications.

**Exercise 2.18.** Think of the keywords and heading for the article.

**Exercise 2.19.** Choose the correct form of the verb in *italics*.

**Exercise 2.20.** Fill in the appropriate link words (you will not need all of them): *to start with, thereafter, in other words, later, next, as a result, hence, however, in conclusion.*

**Exercise 2.21.** Complete the definitions using the word from the text.

- 1. \_\_\_\_\_ a program or piece of software designed to fulfil a particular purpose.
- 2. \_\_\_\_\_ the quality of being new, original, or unusual.
- 3. \_\_\_\_\_ an error in a computer program or system.
- 4. \_\_\_\_\_ to keep from happening, especially by taking precautionary action.
- 5. \_\_\_\_\_ to make or become greater in size, degree, frequency, etc.; grow or expand.

Grammar focus: Impersonal constructions

1. <b>We believe</b> that our study and techniques have the potential to help developers increase the quality of Android app.	A. <b>It is believed</b> that our study and techniques have the potential to help developers increase the quality of Android applications. B. Our study and techniques <b>are believed</b> to have the potential to help developers increase the quality of Android app.
---	---

**Exercise 2.22.** Rewrite the sentences using impersonal constructions.

- 1. We show how these techniques helped to re-discover existing bugs.  
A. \_\_\_\_\_  
B. \_\_\_\_\_
- 2. We believe that our framework can be easily extended to find a broader range of bugs in Android applications.  
A. \_\_\_\_\_  
B. \_\_\_\_\_
- 3. A study by Maji et al. [27] has found that Android applications can have defect densities orders of magnitude higher than the OS.

- A. \_\_\_\_\_  
 B. \_\_\_\_\_
4. In this paper, we show how to construct an effective test automation approach for addressing such bugs.  
 A. \_\_\_\_\_  
 B. \_\_\_\_\_
5. We have found that the remaining bugs are Android-specific.  
 A. \_\_\_\_\_  
 B. \_\_\_\_\_
6. From the results, we found that 25 out of the 50 apps were not fully covered.  
 A. \_\_\_\_\_  
 B. \_\_\_\_\_

**Exercise 2.23.** The sentences given below contain an extra word. Cross out the extra word where necessary.

1. We present an approach for an automating the testing process for Android applications, with a focus on GUI bugs.

2. Our study has found that GUI bugs are quite more numerous.

3. Our study and techniques have the potential as to help developers increase the quality of Android applications.

4. Many tools and techniques exist for automating the testing of and mature, well-established applications, such as desktop or server programs.

5. In summary, our work is tackles the challenges of verifying mobile applications and makes two contributions.

**Exercise 2.24.** Boost your vocabulary with the words from the Abstract.

Noun	Verb	Adjective	Adverb
reliability	rely	computational	increasingly
verification	conduct	numerous	
novelty	affect	certain	
paradigm	feed		
frequency			
approach			

## Part D

**Exercise 2.25.** Read and analyze the Abstract below.

### ORM/NIAM Object-Role Modeling

#### Abstract

Object-Role Modeling (ORM) (*to be*) a method for modeling and querying an information system at the conceptual level, and mapping between conceptual and logical (e.g. relational) levels. ORM (*to come*) in various flavors, including NIAM (Natural language Information Analysis Method). This article (*to provide*) an overview of ORM, and (*to note*) its advantages over Entity Relationship and traditional Object-Oriented modeling.

**Exercise 2.26.** Put the verbs in brackets in the correct tense form

**Exercise 2.27.** What is the weakness of this Abstract?

**Exercise 2.28.** Explain the meaning of the highlighted word combinations in your own words.

**Verbs that are frequently used in IT research papers**

1. To model	To use (a system, procedure, etc.) as an example to follow or imitate.
2. To query	to request data or information from a database table or combination of tables.
3. To map	to take several things and then somehow associate each of them with another thing.
4. To include	to take in or comprise as a part of a whole.
5. To provide	To make available for use; supply.

**Exercise 2.29.** Complete the sentences with the verbs from the table in the appropriate form.

1. The precision ... the percentage of the recovered pattern instances that are correct design pattern instances,

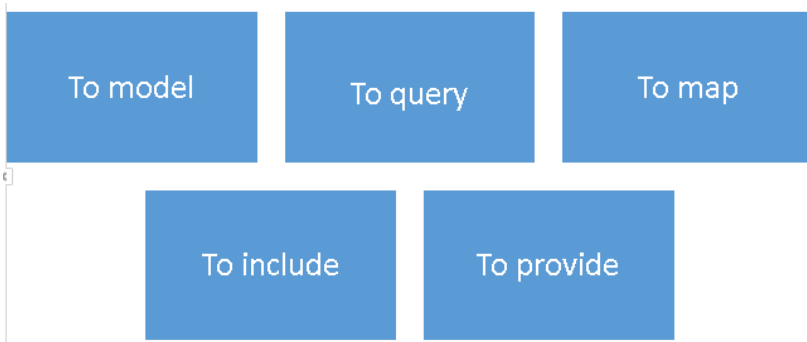
2. These activities are concerned with taking the functional requirements and ... them to a new notation or form.

3. Understand and ... relevant aspects of the process for creating OO designs.

4. The design ... information that, while perhaps true, does not apply to this domain and should not be ... in the design.

5. This makes it possible to introspect and ... all the elements of the UI.

**Exercise 2.30.** Tell your partner/group what it is possible



## Part E

**Exercise 2.31.** Read and analyze the Abstract below.

Design pattern recovery through visual language parsing and source code analysis

Andrea De Lucia, Vincenzo Deufemia, Carmine Gravino \*, Michele Risi

Dipartimento di Matematica e Informatica, Universita di Salerno,  
Via Ponte Don Melillo, 84084 Fisciano (SA), Italy

article info

Article history:

Received 23 October 2008

Received in revised form 5 February 2009

Accepted 5 February 2009

Available online 20 February 2009

Keywords:

Reverse engineering

Design pattern recovery

LR-based parsing

Source code analysis



## Abstract

*In this paper/here* we (to propose) an approach for recovering structural design patterns from object-oriented source code. The recovery process (to organize) in two phases. In the first phase, the design pattern instances (to identify) at a coarse-grained level by considering the design structure only *but /and/*exploiting a parsing technique used for visual language recognition. *Herein/then*, the identified candidate patterns (to validate) by a fine-grained source code analysis phase. The recognition process (to support) by a tool, *namely/for example* *design pattern recovery environment*, which allowed us to assess the retrieval effectiveness of the proposed approach on six public-domain programs and libraries.

**Exercise 2.32.** Choose the appropriate link words.

**Exercise 2.33.** Put the verbs in brackets in the correct tense form.

**Exercise 2.34.** Match the terms with their definitions

1. A source code	A. a distinct period or stage in a series of events or a process of change or development.
2. A design pattern	B. sensing and encoding of printed or written data by a machine.
3. A process	C. a description of a set of interacting classes that provide a framework for a solution to a generalized problem in a specific context or environment.
4. Recognition	D. a way of dealing with a situation or problem.
5. An approach	E. the process of finding and bringing back something.
6. A phase	F. a text listing of commands to be compiled or assembled into an executable computer program.
7. Retrieval	G. a series of actions which are carried out in order to achieve a particular result.

**Exercise 2.35.** Look up the words with the opposite meaning in the dictionary. Make up the sentences of your own with the words you will find.

**To propose**, put forward, suggest, offer, come up with – .....

1. ....

**To identify**, associate, link, connect, relate, draw a parallel with – .....

2. ....

**To validate**, authenticate, confirm, verify – .....

3. ....

**To support**, help, aid, assist, contribute to – .....

4. ....

**Grammar focus: to allow, enable, permit, encourage, recommend**

<p>These verbs take the to-infinitive when they are followed by the object. They take the – ing form when they are not followed by the object.</p>	<p><b>Allow smb to do smth</b>  <b>Allow doing smth</b>  <b>Allow smth to be done</b></p>
--	---

**Exercise 2.36.** Rewrite the sentences in two possible ways.

1. The recognition process allowed us to assess the retrieval effectiveness of the proposed approach on six public-domain programs and libraries.

A. ....

B. ....

2. This will allow you run the generated test suite against any compiled version of the AUT.

A. ....

B. ....

3. This pause allowed the application to finish the computations before the next random event was triggered.

A. \_\_\_\_\_

B. \_\_\_\_\_

4. The testing tools don't allow the tester to provide custom data for the authentication forms.

A. \_\_\_\_\_

B. \_\_\_\_\_

5. Eye tracking allows us to identify patterns that even the users do not consciously see.

A. \_\_\_\_\_

B. \_\_\_\_\_

## **PART F**

**Exercise 2.37.** Read the Introduction below.

Автоматизация выделения вариантов использования  
из SADT – диаграмм.

О.А. Змеев, А.Ю. Малиновский

### **Введение**

В настоящее время развитие программного обеспечения идет в сторону усложнения и укрупнения разрабатываемых информационных систем. Эта тенденция связана, с одной стороны, с тем, что современные компьютеры становятся все мощнее, и это позволяет их пользователям предъявлять повышенные требования к кругу задач, которые решаются с их помощью. В качестве другой причины можно указать все возрастающее использование технологий Интернет, которые за чрезвычайно короткий промежуток времени прошли путь от

простого форматированного текста к полнофункциональным мультимедийным приложениям. С другой стороны, существует другая тенденция, согласно которой время, отпущенное на разработку программного обеспечения (ПО), существенно сокращается. В этих условиях одной из центральных проблем индустрии ПО становится проблема организации процесса разработки ПО.

В последнее время в индустрии разработки ПО часто используются процессы, основанные на методологии Унифицированного процесса разработки программного обеспечения (UP). К числу таких процессов относится RUP (RationalUnifiedProcess) [1], появившийся в 1995 году в результате слияния процессов RationalApproach и ObjectoryProcess 3.8, процесс ICONIX [2, 3], разработанный Д. Розенбергом, к числу таких процессов достаточно большое число авторов относит и процесс XP [4]. Полностью теоретические концепции, лежащие в основе UP, изложены в [5], в [6] приведен краткий обзор основных идей, на которых базируется процесс.

Одной из ключевых концепций UP является понятие *варианта использования (usecase)*. Обычно UP характеризуется как управляемый вариантами использования, это означает, что команда разработчиков применяет варианты использования для выполнения полного комплекса работ по проекту, начиная с начального сбора информации и заканчивая тестированием готовой программной системы. К сожалению, вопросы, связанные с формированием модели вариантов использования для построения программных систем, на сегодня являются наиболее сложной и наименее формализованной частью UP.

К тому же достаточно часто проекты, связанные с разработкой и внедрением различного рода информационных систем, происходят в рамках процедуры реинжиниринга бизнес-процессов организации клиента. Ключевыми артефактами этой процедуры являются так называемые модели «IT IS» и «TO BE», для получения которых бизнес-аналитики чаще всего используют методологию SADT [7, 8].

В настоящей работе предлагается подход, позволяющий использовать для построения первого приближения модели вариантов использования диаграммы, полученные в результате SADT-моделирования. Если методология SADT применялась для разработки функциональной модели целевой организации в целом, то предлагаемый подход позволяет построить первое приближение для модели деловых (бизнес) вариантов использования. Если соответствующая SADT-модель декомпозирует функциональность целевой системы, то в качестве результата может быть получено первое приближение модели вариантов использования в обычном смысле. Такой подход, на наш взгляд, с одной стороны, увеличивает степень повторного использования артефактов процесса реинжиниринга, а с другой – позволяет получить первое приближение модели вариантов использования более высокого качества и за более короткие сроки.

**Exercise 2.38.** Make the list of keywords in English.

**Exercise 2.39.** Write the Abstract of the paper in English.

**Exercise 2.40.** Render the text in English using the appropriate link words.

**Exercise 2.41.** Translate the following sentences from Russian into English.

1. В этой статье описывается метод ...
2. Эффективность метода была продемонстрирована ...
3. Предложен алгоритм для автоматизированной идентификации возможностей перестройки кода.
4. Предложенный метод для автоматизированного рефакторинга вносит вклад в ...
5. Экспериментальные результаты показывают, что предложенная нами схема достигает гораздо лучших характеристик, чем существующие схемы.
6. Шаблоны дизайна широко используются дизайнерами и разработчиками для построения сложных систем.
7. Поскольку информационные системы обычно реализуются на DBMS в основе которой лежит некая модель логических данных, то объектно-ориентированное моделирование включает процедуры для отображения между концептуальным и логическим уровнями.
8. Для правильности, ясности и применимости информационные системы точно определяются прежде всего на концептуальном уровне
9. Визуальная криптография – это особый тип техники шифрования ...
10. Каждый пиксель оцифрованного цветного изображения, состоящего из 32 битов, делится на 4 части, а именно: альфа, красный, зеленый и голубой, при этом каждый состоит из 8 битов.

## **Revision**

What do you need to know when writing an Abstract?

### **Self-study activity**

1. Watch the video: How to Write an Abstract for a Research Paper

<https://www.youtube.com/watch?v=JMEnRBss6V4&t=57s>

2. Answer the questions:

What information does the abstract include?

How many paragraphs does the abstract consist of?

Are there any keywords in the abstract?

3. Write the draft of the abstract for the research paper of your own.

**The phrases that may help you:**

*X is recognized as being / believed to be / widely considered to be the most important ...*

*It is well known / common knowledge that X is ...*

*Xs are attracting considerable / widespread interest due to ...*

*Until now / For many years / Since 1993 Xs have been considered as ...*

*X has received much attention in the last two years / in the past decade / over the last two decades ... For the past five years...*

*In this report / paper/study we ...*

*This paper outlines / proposes / describes / presents a new approach to...*

*This paper seeks to address / focuses on / discusses / ...*

*This paper is an overview of / a review of / a report on / a preliminary attempt to ...*

*The aim of our work / research / study was to further extend / widen / broaden current knowledge of ...*

*The aim of this study is to evaluate / validate / determine ...*

*With this in mind / Within the framework of these criteria / In this context we tried to ...*

*We believe that we have found / developed / discovered / designed...*

*Future work will concentrate on / focus on / explore / investigate / look into*

*Further studies, which take X into account, will need to be undertaken / performed to ...*

### 1.3. THE STRUCTURE OF THE INTRODUCTION

#### *Focus on theory*

In the Introduction, the background knowledge is presented. The reader finds the tools for understanding the meaning and motivation of the research carried out by the authors.

In the Introduction, the author introduces:

- The description of the problem or establishing a context;
- The literature review;
- Research gaps in a certain scientific field;
- The gap the author is going to fill;
- The definition of the purpose;
- The results the author has arrived at;
- The structure of the rest of the paper.

#### *The use of tenses*

The present simple is generally used at the beginning of the Introduction in order to describe well-known facts. Then it is reasonable to use the Present Perfect to show how long the problem has been studied. To introduce the completed investigations the Past Simple is used. At the end of the Introduction the Present Simple is usually used to outline the structure of the paper.



## PART A

**Exercise 3.1.** Read and analyze the Introduction using the information given above. Put the verbs in brackets into the correct form.

### Object-Role Modeling (ORM/NIAM)

Terry Halpin

Microsoft Corporation, USA

#### 1 Introduction

##### 1.1 *ORM: what is it and why use it?*

*Object-Role Modeling* (ORM) is primarily a method for modeling and querying an information system at the conceptual level. In Europe, the method ... often ... **(call)** NIAM (Natural language Information Analysis Method). Since information systems ... typically ... **(implement)** on a DBMS that is based on some logical data model (e.g. relational, object-relational, hierarchic), ORM ... **(include)** procedures for mapping between conceptual and logical levels. Although various ORM extensions ... **(propose)** for process and event modeling, the focus of ORM is on data modeling, since the data perspective is the most stable and it ... **(provide)** a formal foundation on which operations can be defined.

For correctness, clarity and adaptability, information systems ... best ... **(specify)** first at the conceptual level, using concepts and language that people can readily understand. Analysis and design ... **(involve)** building a formal model of the application area or *universe of discourse* (UoD). To do this properly ... **(require)** a good understanding of the UoD and a means of specifying this understanding in a clear, unambiguous way. Object-Role Modeling. **(simplify)** this process by using natural language, as well as intuitive

diagrams that can be populated with examples, and by expressing the information in terms of elementary relationships.

ORM ... **(so-call)** because it pictures the world in terms of *objects* (entities or values) that play *roles* (parts in relationships). For example, you are now playing the role of reading, and this paper is playing the role of being read. In contrast to other modeling techniques such as Entity-Relationship (ER) and Object-Oriented (OO) approaches, ORM makes no explicit use of *attributes*. For example, instead of using `countryBorn` as an attribute of `Person`, we ... **(use)** the relationship type `Person` was born in `Country`. This ... **(have)** many important advantages. Firstly, ORM models and queries are more stable (attributes may evolve into entities or relationships). For example, if we decide to later record the population of a country, then our `countryBorn` attribute needs to be reformulated as a relationship. Secondly, ORM models may ... conveniently ... **(populate)** with multiple instances (attributes make this too awkward). Thirdly, ORM is more uniform (e.g. we don't need a separate notation for applying the same constraint to an attribute rather than a relationship).

ORM is typically more expressive than ER or OO. Its role-based notation makes it easy to specify a wide variety of constraints, and its object types ... **(reveal)** the semantic domains that bind a schema together. One benefit of this is that conceptual queries may now ... **(formulate)** in terms of schema paths, where moving from one role through an object type to another role amounts to a conceptual join (see later).

Unlike ORM or ER, popular OO models often ... **(duplicate)** information by wrapping facts up into pairs of inverse attributes in different objects. Moreover, OO notations have weak support for constraints (e.g. a constraint might have to be duplicated in different

objects, or even ignored). Unfortunately, OO models are less stable than even ER models when the UoD ... **(evolve)**. For such reasons, OO models should ... **(use)** only for implementation, not for analysis.

Although the detailed picture provided by ORM is desirable in developing and transforming a model, for summary purposes it is useful to hide or compress the display of much of this detail. Various abstraction mechanisms ... **(exist)** for doing this [e.g. CHP96]. If desired, ER and OO diagrams can also ... **(use)** for providing compact summaries, and ... best ... **(develop)** as views of ORM diagrams. For a simple discussion illustrating the points in this section, see [Hal96].

The rest of this article ... **(provide)** a brief history of ORM, summarizes the ORM notation, ... **(illustrate)** the conceptual design and relational mapping procedures, and ... **(mention)** some recent extensions before concluding.

**Exercise 3.2.** Answer the questions.

1. What problem does the author describe?
2. Is there the literature review?
3. Which gap is the author going to fill in the research?
4. How does the author define the purpose of the research?
5. Are the results the author has arrived at given?
6. Is there the structure of the rest of the paper?

**Exercise 3.3.** Underline the link words that are used in the paper and put them into the correct column of the table below.

<i>Reason</i>	<i>Contrast</i>	<i>Listing points</i>	<i>Positive addition</i>	<i>Example</i>

**Exercise 3.4.** Fill in the prepositions *of* (2) *on* (2) *to* *in*(2) to make word combinations complete.

1. the focus of ORM is ... data modeling
2. in contrast ... other modeling techniques
3. expressing the information ... terms ... elementary relationships
4. based ... some logical data model
5. instead ... using countryBorn as an attribute of Person
6. the detailed picture is desirable ... developing a model

**Exercise 3.5.** Match the terms with their definitions.

1. Adaptability	A. a quality or characteristic that someone or something has
2. A procedure	B. the act of putting a plan into action or of starting to use something
3. A technique	C. a set of instructions in a computer program that does a particular task
4. An attribute	D. a general concept or idea, rather than something concrete or tangible
5. An implementation	E. the capacity to be modified for a new use or purpose
6. An abstraction	F. a way of doing an activity that needs skill

**Exercise 3.6.** Complete the sentences with the words from the table.

1. In this paper we present a new ... able to recover behavioral design pattern instances which combines static analysis with dynamic analysis.
2. We want to thank Marco Biscione for his contribution in the ... of the design pattern recovery tool.

3 Table 3 summarizes the size of this design by reporting for each class the number of ...

4. The models of ... are the models of the important information in an artifact, and how it is organized.

5. We can also try to reduce the time needed for encryption and decryption ...

6. This kind of data awareness improves the modularity and mobility of system components and hence system scalability and ...

**Grammar focus: *that or which* in the relative clauses**

<b>THAT</b>	<ul style="list-style-type: none"><li>• refers to people and objects</li><li>• is never used after a comma or preposition</li></ul>
<b>WHICH</b>	<ul style="list-style-type: none"><li>• refers to objects</li><li>• is used after a comma or preposition</li></ul>

**Exercise 3.7.** Read the sentences with the relative clauses. Analyze the use of relative pronouns that and which. Can which be replaced by that?

1. Since information systems are typically implemented on a DBMS ***that*** is based on some logical data model (e.g. relational, object-relational, hierarchic), ORM includes procedures for mapping between conceptual and logical levels.

2. The course aimed to teach software engineering principles, ***which*** were required to be applied to the development of an application over the course of the semester.

3. ORM is so-called because it pictures the world in terms of objects (entities or values) ***that*** play roles (parts in relationships).

4. Its role-based notation makes it easy to specify a wide variety of constraints, and its object types reveal the semantic domains ***that*** bind a schema together.

5. We also compare against a recent method *that* combines two network streams for better performance.

6. Humanoid performed a sequence of meaningful actions, *which* was opposite to Android Monkey’s inability to test new core functionality

7. We propose an inexpensive hard-identity mining scheme *that* helps learn better features.

8. There are apps for *which* all the tools, including the best-performed tool, achieved shallow coverage, i.e., lower than 5%.

9. To demonstrate the effectiveness of our approach, in Section 5 we present an evaluation on the open source applications *that* form the object of our bug study.

10. Error diffusion is a type of halftoning technique in *which* the quantization error of a pixel is distributed to neighboring pixels

**Exercise 3.8.** Boost your vocabulary with the words from the Introduction.

Noun	Verb	Adjective	Adverb
query	implement	hierarchic	conveniently
adaptability	propose	conceptual	primarily
variety	specify	compact	
constraint	evolve	unambiguous	
path	reveal	awkward	
	simplify	explicit	
	conclude	separate	
	populate	desirable	
	map	various	
	bind	recent	

## PART B

**Exercise 3.9.** Read and analyze the Introduction given below.

### Automating GUI Testing for Android Applications

Cuixiong Hu Iulian Neamtii

Department of Computer Science and Engineering

University of California, Riverside, CA, USA

{huc,neamtii}@cs.ucr.edu

#### 1. INTRODUCTION

Smartphones are becoming pervasive, with more than 195 million sold worldwide in the first three quarters of 2010 alone [23, 22, 24]. A major draw of any smartphone is its ability to run applications, thus users are increasingly relying on smartphones for computing needs, rather than using laptops or desktops. This leads to an increasing impetus for ensuring the reliability of mobile applications. Reliability is particularly important for sensitive mobile applications such as online banking, business management, health care, or military domains.

In *this paper/this research* we focus on ensuring the reliability of mobile applications running on the Google Android platform. *By /According to* Fall 2010 reports, Android is the second most popular mobile OS, surpassing BlackBerry and iPhone OS, and will be tied for number one with Nokia's Symbian by 2014 [29, 24]; Android is *in fact/really* the only mobile OS platform to gain market share since Q4'09 [23, 22, 24]. The Android ecosystem includes the Android Market, which currently lists more than 220,000

applications, 12,316 of which were added in December 2010 alone, and an estimated 2.6 billion downloads [17].

Many tools and techniques exist for automating the testing of mature, well-established applications, *like/such as* desktop or server programs. *However/though*, the physical constraints of mobile devices (e.g., low-power CPU, small memory, small display), as well as developers' unfamiliarity with mobile platforms (*due to/because* of their novelty), make mobile applications prone to new kinds of bugs. *For instance/ For example*, an Android application is structured around activities (GUI windows), broadcast receivers, services and content providers; this is different from standard server applications, or from an event-based system used in a desktop GUI application. The tendency of mobile applications to have bugs is evidenced by their high defect density: a study by Maji et al. [27] has found that Android applications can have defect densities orders of magnitude higher than the OS.

In *this paper /this research* we aim to bring novel, Android-specific classes of bugs to light, and show how to construct an effective test automation approach for addressing such bugs, *especially/special* GUI bugs, and ensuring the reliability of Android applications. *First/to start with*, we conduct a bug collection and categorization on 10 popular open source Android applications (Section 2). We found that, while bugs related to application logic are still present, the remaining bugs are Android-specific, i.e., *thanks/ due* to the activity- and event-based nature of Android applications. We categorized all confirmed bugs in the bug database based on our observations. To detect and fix these categories of bugs, we employ an automated test approach (Section 4). Our approach uses a combination of techniques. *First/to begin with*, we employ test and event generators to construct test cases and sequences of events. We



then run these test cases (and feed the events, respectively) to the application. *As soon as /Once* a test case is running, we record detailed information about the application in the system log file; after each test case run, we perform a log file analysis to detect potential bugs. To demonstrate the effectiveness of our approach, in Section 5 we present an evaluation on the open source applications that form the object of our bug study. We generated test cases for all projects used in the bug study and compared bugs we found with bugs reported by users. We detected most bugs reported, and found new bugs which have never been reported.

***In summary/in conclusion***, our work tackles the challenges of verifying mobile applications and makes two contributions:

1. A bug study and categorization of Android-specific bugs that shows an important number of Android bugs manifest themselves in a unique way that is different from traditional, e.g., desktop/server application bugs.
2. An effective approach for detecting Android GUI bugs, based on a combination of test case and event generation with runtime monitoring and log file analysis.

***Exercise 3.10.*** Choose the appropriate link words in the text given above. Remember that sometimes both of them are possible.

***Exercise 3.11.*** Mark the following as true or false.

1. A major draw of any smartphone is its ability to run applications, thus users are increasingly relying on smartphones for computing needs, rather than using laptops or desktops.
2. Reliability is the least important for sensitive mobile applications such as online banking, business management, health care, or military domains.

3. Android is the first most popular mobile OS.
4. Our approach uses a combination of techniques.
5. In Section 3 we present an evaluation on the open source applications that form the object of our bug study

**Exercise 3.12.** Complete the definition using the correct word from the text. Make up the sentences of your own with the words you filled in.

1. \_\_\_\_\_ *(especially of an unwelcome influence or physical effect) spreading widely throughout an area or a group of people.*

2. \_\_\_\_\_ *the force or energy with which a body moves; something that makes a process or activity happen or happen more quickly.*

3. \_\_\_\_\_ *the use or introduction of automatic equipment in a manufacturing or other process or facility.*

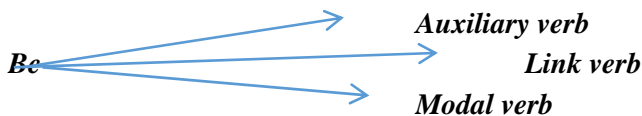
4. \_\_\_\_\_ *make use of.*

5. \_\_\_\_\_ *make determined efforts to deal with (a problem or difficult task).*

**Exercise 3.13.** Fill in the prepositions of (2) from to with on to make word combinations (phrases) complete.

- A major draw ... any smartphone
- This leads ... an increasing impetus
- Mobile applications running ... the Google Android platform
- 12,316 ... which were added
- Developers' unfamiliarity ... mobile platforms
- This is different ... standard server.

Grammar focus: different functions of the verb *to be*



**Exercise 3.14.** How do you think the highlighted words differ?

1. We detected most bugs reported, and found new bugs which **have never been** reported.

2. Android **is** the second most popular mobile OS, surpassing BlackBerry and iPhone OS, and **will be** tied for number one with Nokia's Symbian by 2014.

3. Smartphones **are** becoming pervasive, with more than 195 million sold worldwide in the first three quarters of 2010 alone.

4. A major draw of any smartphone **is** its ability to run applications, thus users **are** increasingly relying on smartphones for computing needs, rather than using laptops or desktops.

5. This method can further be extended to (n, n) visual cryptography scheme.

6. The advantage of visual cryptography is that no complex mathematical computation is needed to decrypt the message hence the decryption can be performed by any human visual system.

7. In Visual Cryptography the Image is divided into parts called shares and then they are distributed to the participants.

8. The objective of this paper is to give a brief introduction to distributed object technology and an overview of security features available in Microsoft.NET and CORBA.

9. In this method two levels of security are involved.

10. We are to continue this work.

**Exercise 3.15.** Boost your vocabulary with the words from the Introduction.

Noun	Verb	Adjective	Adverb
draw	surpass	pervasive	increasingly
impetus	estimate	mature	particularly
reliability	confirm	prone	respectively
constraint	employ	novel	
density	tackle	unique	
sequence	verify		
evaluation			
test cases			
challenge			

**PART C**

**Exercise 3.16.** Read and analyze the Introduction given below.

**Simulating Non Stationary Operators in Search Algorithm**

Adrien Goeffon

LERIA, University of Angers (France)

Frederic Lardeux

LERIA, University of Angers (France)

Frederic Saubion

LERIA, University of Angers (France)

# 1. INTRODUCTION

## 1. Contributions

Our simulation model is used to compare the different performances of operator selection policies and clearly identify their ability to adapt to such specific operators behaviours and can be used as a surrogate operator model when designing new adaptive search algorithms. Hence, the general description of operator based search algorithms may be helpful in this design process when the user has to precisely identify the components and performance criteria that are used in the adaptive process. The experimental study provides interesting results on the respective behaviours of operator selection policies when faced to such non stationary search scenarios. Considered as a multiarmed bandit problem, our model corresponds to a specific restless bandit problem that could be used to model different real applications as soon as the efficiency of a given action decreases according to successive frequent uses. For instance, such reinforcement learning techniques are used to schedule online advertisement display on web pages. Our model could be pertinent in this context since it may be clear that the relevance of an advertisement decreases if it is too much shown to the same user. Other cases of such decrease repeated actions may actually be observed in various application domains.

2. Selecting the most suitable operators in a search algorithm when solving optimization problems is an active research area (Eiben et al., 2007; Lobo et al., 2007). Given an optimization problem, a search algorithm mainly consists in applying basic solving operators — heuristics — in order to explore and exploit the search space for retrieving solutions. The choice of the successive operators along the search process is often driven by means September 8, 2014 of parameters. The improvement of the performance of the algorithm

thus relies on an adequate setting of these parameters. Unfortunately, according to the underlying intuitions provided by the No Free Lunch theorems for optimization (Wolpert and Macready, 1997), this optimal policy may strongly depend on the instances of the problems to be solved.

### 3. Organization of the paper

In section 2, we describe optimization algorithm that are based on applications of basic search operators. We also define the problem of designing the best possible operator selection policy and show its relationship with multiarmed bandit problems. Section 3 is dedicated to review different operator selection policies. Section 4 presents our model for simulating non stationary operators. Experiments are presented in section 5.

### 4. Motivations

In this paper, we propose an alternative model for simulating search operators whose behaviour often change continuously during the search. In these scenarios, the performance of the operators decreases when they are applied. This is motivated by the fact that operators for optimization problems are often roughly classified into exploitation operators and exploration operators. Exploitation operators aim at focusing on the evaluation of the visited configurations of the search space in order to converge quickly to a local optimum. Exploration operators aim at diversifying the search trajectory by visiting sparse areas of the search space. Unfortunately, it is not possible to always exploit nor explore the search space. For instance, it is unlikely that an exploitation operator will always improve a configuration and find directly an optimal solution (except for simple problems). Therefore, decreasing performance may be observed along the search as well as changing behaviours of operators.

5. Initial parameters setting can be achieved by automated tuning algorithms (Hutter et al., 2009; Nannen et al., 2008). Nevertheless, the values of the parameters may require more continuous control (Fialho, 2010) and should rather not be fixed during the whole search process. Adaptive operator selection is strongly related to reinforcement learning problems, and especially to multi-armed bandit problems (Fialho et al., 2008; Costa et al., 2008). Various methods for managing the famous exploration vs. exploitation balance in search heuristics have been investigated in the literature; see for instance (Maturana et al., 2009; Lobo et al., 2007; Thierens, 2005). The performance of adaptive selection policies depends on the characteristics of the problem's search space, as well as on the specificities of the search operators. Therefore different families of practical problems have been handled, but also more abstract operators models in order to provide more general and comprehensive testing frameworks as in Thierens (2005) and Costa et al. (2008), taking into account changes in the operators behaviours.

**Exercise 3.17.** Look through the Introduction. Put the paragraphs in the logical order. Underline link words.

**Exercise 3.18.** Find the words in the text with the same meaning.

- a) *exactly* \_\_\_\_\_
- b) *useful* \_\_\_\_\_
- c) *element, ingredient, constituent* \_\_\_\_\_
- d) *to utilize, to use* \_\_\_\_\_
- e) *to investigate* \_\_\_\_\_
- f) *approximately* \_\_\_\_\_
- g) *complete, exhaustive, thorough* \_\_\_\_\_

**Exercise 3.19.** Fill in the prepositions for into in (2) on to(3) to make word combinations (phrases) complete.

1. their ability to adapt ... such specific operators behaviours
2. interesting results ... the respective behaviours
3. when faced ... such non stationary search scenarios
4. according ... successive frequent uses
5. may actually be observed ... various application domains.
6. a search algorithm mainly consists ... applying basic solving operators
7. in order to explore and exploit the search space ... retrieving solutions.
8. taking ... account changes in the operators behaviors

**Exercise 3.20.** Match the terms with their definitions.

1.The testing framework	A. a document for the World Wide Web that is identified by a unique uniform resource locator (URL)
2. Heuristics	B. allow the development of algorithms to learn the solutions to the optimal control problems for dynamic systems that are described by difference equations.
3. An application domain	C. is a set of guidelines or rules used for creating and designing test cases.
4. The optimization problem	D. the step-by-step procedure used to locate specific data among a collection of data.
5.A web page	E. a technique designed for solving a problem more quickly when classic methods are too slow or for finding an approximate solution when classic methods fail to find any exact solution.
6. Reinforcement learning techniques	F. the segment of reality for which a software system is developed
7. The search algorithm	G. the problem of finding the best solution from all feasible solutions



Grammar focus: how to express *purpose*

<b>Purpose</b>	1.(in order) + to Infinitive 2. for + Gerund
----------------	---

*Exercise 3.21.* What do the highlighted phrases express?

1. Our simulation model is used to compare the different performances of operator selection policies and clearly identify their ability to adapt to such specific operators behaviors and can be used as a surrogate operator model when designing new adaptive search algorithms.

2. Considered as a multiarmed bandit problem, our model corresponds to a specific restless bandit problem that could be used to model different real applications as soon as the efficiency of a given action decreases according to successive frequent uses.

3. Section 4 presents our model for simulating non stationary operators.

4. In this paper, we propose an alternative model for simulating search operators whose behaviour often change continuously during the search.

5. Therefore different families of practical problems have been handled, but also more abstract operators models in order to provide more general and comprehensive testing frameworks as in Thierens (2005) and Costa et al. (2008), taking into account changes in the operators behaviours.

6. Exploitation operators aim at focusing on the evaluation of the visited configurations of the search space in order to converge quickly to a local optimum.

7. Visual cryptography is a new technique which is used to secure the visual content such as texts or images.

8. They represent a useful technique in forward engineering since they allow reusing successful practices, to improve communication between designers, and to share knowledge between software engineers.

9. We first conducted a bug mining study to understand the nature and frequency of bugs affecting Android applications.

10. To detect and fix these categories of bugs, we employ an automated test approach.

**Exercise 3.22.** What does *it* mean in these sentences?

1. *It* may be clear that the relevance of an advertisement decreases if *it* is too much shown to the same user.

2. *It* is necessary to modify the current solution due to various changes in the environment.

3. *It* is a  $(n + 1)$ -dimensional search-space.

4. When the changes have to be detected first, the information about them can be given to the system from outside or *it* can deduced on the basis of behavior of system components.

5. Changes which are continuous in time make the environment at least a little different every time we measure *it*.

**Exercise 3.23.** Boost your vocabulary with the words from the Introduction.

Noun	Verb	Adjective	Adverb
surrogate	explore	respective	precisely
performance	exploit	successive	unfortunately
efficiency	converge	pertinent	roughly
reinforcement	diversify	adequate	
relevance		sparse	
heuristics		comprehensive	

## PART D

**Exercise 3.24.** Read the Abstract and Introduction given below. Compare the information given in the Abstract and Introduction. Choose the correct form of the infinitive.

### **Design pattern recovery through visual language parsing and source code analysis**

Andrea De Lucia, Vincenzo Deufemia, Carmine Gravino \*,  
Michele Risi

Dipartimento di Matematica e Informatica, Università di Salerno,  
Via Ponte Don Melillo, 84084 Fisciano (SA), Italy

#### **a r t i c l e i n f o**

Article history:

Received 23 October 2008

Received in revised form 5 February 2009

Accepted 5 February 2009

Available online 20 February 2009

**Keywords:** Reverse engineering, Design pattern recovery, LR-based parsing, Source code analysis

#### **a b s t r a c t**

In this paper we propose an approach for recovering structural design patterns from object-oriented source code. The recovery process is organized in two phases. In the first phase, the design pattern instances are identified at a coarse-grained level by considering the design structure only and exploiting a parsing technique used for visual language recognition. **Then**, the identified candidate patterns are

validated by a fine-grained source code analysis phase. The recognition process is supported by a tool, namely *design pattern recovery environment*, which allowed us 1. *to assess/to have been assessed* the retrieval effectiveness of the proposed approach on six public-domain programs and libraries.

\_ 2009 Elsevier Inc. All rights reserved.

## 1. Introduction

A design pattern can be seen as a set of classes, related through inheritances, aggregations and delegations, which represents a partial solution to a common non-trivial design problem (Gamma et al., 1995). Design patterns are widely used 2. *to separate/to have been separating* an interface from the different possible implementations, 3. *to be wrapping/to wrap* legacy systems, 4. *to encapsulate/to be encapsulated* command requests, 5. *to use/to have been used* different platforms, and so on (Gamma et al., 1995). They represent a useful technique in forward engineering since they allow reusing successful practices, to improve communication between designers, and to share knowledge between software engineers. **However**, patterns can also 6. *be used/be using* for reverse engineering OO software systems in order 7. *to capture/to have been captured* relevant information on the design and code, and improve program understanding (Antoniol et al., 2001; Brown, 1996; Niere et al., 2002; Shull et al., 1996; Tsantalis et al., 2006a). **As a matter of fact**, the use of patterns during the design phases affects the corresponding code, and the extraction of design pattern information from design and code can help the comprehension of the adopted solution for a system. This information can be used 8. *to be highlighted /to highlight* wished properties of the design model, which can 9. *be reused/ reuse* whenever a similar problem is encountered. **Indeed**, as also

highlighted in Antoniol et al. (2001) when a software system has been designed using documented and well-known design patterns they can exhibit good properties such as modularity, separation of concerns, and ease of extension. **Moreover**, the information on the recovered design patterns can improve the system documentation and can guide the restructuring of the system. **As a matter of fact**, the recovery of design pattern instances from design documents and corresponding source code can be crucial for the identification of traceability links between different software artifacts in order 10. *to be made/to make* the code easier to maintain and modify. **In particular**, this information can be profitably exploited 11. *to highlight/to have highlighted* the rationale of implemented solutions in order to support and simplify the conceptual modeling of the system 12. *to have been restructured/ to be restructured*. (Antoniol et al., 2001).

According to (Gamma et al., 1995), design patterns are classified as structural, which concentrate on object composition and their relations in the run-time object structures, creational, which address object instantiation issues, and behavioral, which focus on the internal dynamics and object interaction in the system. In this paper we present an approach 13. *to recover/to be recovering* structural design patterns from OO source code, which is based on the use of visual language grammars and parsing techniques. A preliminary analysis is carried out 14. *to have extracted/ to extract* the structural information needed to recover design patterns. **In particular**, the class diagram information, such as the name and type of classes, methods, and fields, inheritance and association relationships, and so on, are stored in a suitable data structure that speeds up the recovery process. The recovery process combines a diagram-level analysis, by using a parser for visual languages, with a source code level analysis. **In particular**, the recovery process is organized in two phases. In the

first phase, design pattern instances are identified based on the design structure only by using a recovery technique based on visual language parsing (Costagliola et al., 2005). The design pattern recovery problem is reduced to the problem of recognizing subsentences in a class diagram, where each subsentence corresponds to a design pattern specified by a grammar. In the second phase the identified candidate patterns are validated by performing a source code analysis, which eliminates false positives and consequently increases the precision (Salton and McGill, 1983) of the recovery approach. 15. ***To validate/to have been validated*** the proposed design pattern recovery approach, we have developed a tool, named Design Pattern Recovery Environment (DPRE), which supports the whole recovery process.

In this paper, we extend the work presented in Costagliola et al. (2005); Costagliola et al. (2006); De Lucia et al. (2007) by:

- presenting a recovery technique supporting design pattern definitions that include multi-level inheritance relationships;
- providing a detailed description of the proposed approach, including the visual parsing phase and the source code analysis phase;
- presenting a classification and an analysis of the design pattern recovery approaches proposed in the literature;
- evaluating the approach and tool on six public-domain software systems and libraries of different size, ranging from 8 to 560 KLOC;
- providing a detailed comparison with related approaches that used the same software systems for the evaluation.

The paper is organized as follows. In Section 2, we describe related work on design pattern recovery. Section 3 presents the proposed design pattern recovery process while Section 4 describes the tool DPRE supporting it. The results of the case studies are reported and discussed in Section 5. Conclusion and future work are given in Section 6.

**Exercise 3.25.** Read the Introduction and state:

- the problem under study,
- the existing solutions (literature),
- the best solution,
- the research gap,
- the goal of research,
- the evaluation of research,
- the structure of the paper.

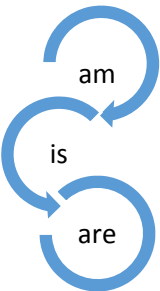
**Exercise 3.26.** Fill in the prepositions *from in of (2) to with for(2)*  
-- to make word combinations (phrases) complete.

1. which represents a partial solution ... a common non-trivial design problem
2. a useful technique ... forward engineering
3. can also be used ... reverse engineering OO software systems
4. the extraction ... design pattern information ... design
5. can be crucial ... the identification ... traceability links between different software artifacts
6. the use of patterns during the design phases affects ... the corresponding code
7. providing a detailed comparison ... related approaches

**Exercise 3.27.** Put the words in the correct order in each sentence.

1. process, The, two, organized, recovery, is, in, phases.
2. case, the, The, discussed, 5, results, are, in, reported, of, studies, Section, and.
3. information, design, the, Moreover, the, on, recovered, patterns, can, the, system, improve, documentation.

4. In, source, recovering, paper, we, approach, structural, for, this, an, design, from, object-oriented, propose, code, patterns.

Grammar focus: Present Passive		
	<p><b>Past</b></p> <p><b>Participle</b></p>	<p>The passive is often used to describe the steps in the process where the action is more important than the agent or the agent is already known.</p>

**Exercise 3.28.** Rewrite the sentences in the Active Voice.

1. In this paper an approach for recovering structural design patterns from object-oriented source code is proposed. \_\_\_\_\_

2. The recovery process is organized in two phases. \_\_\_\_\_

3. Patterns are also used or reverse engineering OO software systems in order to capture relevant information on the design and code, and improve program understanding \_\_\_\_\_



---

4. In the second phase the identified candidate patterns are validated by performing a source code analysis, which eliminates false positives and consequently increases the precision of the recovery approach. \_\_\_\_\_

---

---

5. The paper is organized as follows. \_\_\_\_\_

---

---

6. The results of the case studies are reported and discussed in Section 5. \_\_\_\_\_

---

---

7. Conclusion and future work are given in Section 6. \_\_\_\_\_

---

---

8. Initial parameters setting is achieved by automated tuning algorithms. \_\_\_\_\_

---

---

9. Various methods for managing the famous exploration vs. exploitation balance in search heuristics are investigated in the literature. \_\_\_\_\_

---

---

10. An optimal setting is achieved by an optimal operator selection policy. \_\_\_\_\_

---

**Exercise 3.29.** These three sentences contain errors. Can you spot them yourself?

1. The class diagram information, such as the name and type of classes, methods, and fields, inheritance and association relationships, and so on, are stored in a suitable data structure that slow down the recovery process.

2. A design pattern can be seen as a set of problems, related through inheritances, aggregations and delegations, which represents a partial solution to a common non-trivial design problem.

3. The recovery process combines a diagram-level analysis, by using a library for visual languages, with a source code level analysis.

**Exercise 3.30.** Match the pairs of synonyms.

1. However	a. Furthermore
2. As a matter of fact	b. Especially
3. Indeed	c. Actually
4. Moreover	d. Nevertheless
5. In particular	e. In reality

**Exercise 3.31.** Boost your vocabulary with the words from the Introduction.

Noun	Verb	Adjective	Adverb
recovery	parse	structural	
comprehension	assess	object-oriented	
pattern	encapsulate	fine-grained	
phase	encounter	coarse-grained	
instance		legacy	
environment			
retrieval			
inheritance			
rationale			
instantiation			

## PART E

**Exercise 3.32.** Read and analyze the Abstract and Introduction given below. Compare the information given in the Abstract and Introduction. Choose the correct form of the word in *bold italics*.

**Visual Cryptography Scheme for Color Image Using Random Number  
with Enveloping by Digital Watermarking**

Shyamalendu Kandar<sup>1</sup>, Arnab Maiti<sup>2</sup>, Bibhas Chandra Dhara<sup>3</sup>  
<sup>1,2</sup> Computer Sc. & Engineering  
Haldia Institute of Technology  
Haldia, West Bengal, India  
<sup>3</sup>Department of Information Technology  
Jadavpur University  
Kolkata, West Bengal, India

## Abstract

Visual Cryptography is a special type of encryption technique to obscure image-based secret information which can be decrypted by Human Visual System (HVS). This cryptographic system encrypts the secret image by dividing it into  $n$  number of shares and decryption is done by superimposing a certain number of shares( $k$ ) or more. Simple visual cryptography is insecure because of the decryption process done by human visual system. The secret information can be retrieved by anyone if the person gets at least  $k$  number of shares. Watermarking is a technique to put a signature of the owner within the creation.

In this current work we have proposed Visual Cryptographic Scheme for color images where the divided shares are enveloped in other images using invisible digital watermarking. The shares are generated using Random Number.

**Keywords:** *Visual Cryptography, Digital Watermarking, Random Number.*

## 1. Introduction

Visual cryptography is a cryptographic technique where visual information (Image, text, etc.) gets ***cryptography/encrypted/decrypted/decryption*** in such a way that the decryption can be performed by the human visual system without aid of computers [1]. Like other multimedia components, image is sensed by human. Pixel is the smallest unit ***construction/construct/constructing/constructed*** a digital image. Each pixel of a 32 bit digital color image are divided into four parts, namely Alpha, Red, Green and Blue; each with 8 bits. Alpha part ***representation / represents /representative/represented*** degree of transparency.

A 32 bit sample pixel is represented in the following figure [2] [3].

**11100111**

**11011001**

**11111101**

**00111110**

**Alpha Red**

**Green Blue**

Fig 1: Structure of a 32 bit pixel

Human */visualize/vision /visual/visible* system acts as an OR function. Two transparent objects stacked together, produce transparent object. But *exchange changing/ change / changed* any of them to non-transparent, final objects will be seen non-transparent. In k-n secret sharing visual cryptography scheme an image is divided into n number of shares such that minimum k number of shares is sufficient to reconstruct the image. The division is done by Random Number generator [4].

This *type/typical typist/typed* of visual cryptography technique is insecure as the reconstruction is done by simple OR operation.

To *adding/add/ additive/addition* more security to this scheme we have proposed a technique called digital enveloping. This is nothing but an *extend/ extended/extension/extensive/* invisible digital watermarking technique. Using this technique, the divided shares produced by k-n secret sharing visual cryptography are embedded into the envelope images by LSB *replace /place/ replacement /replaced* [5]. The color change of the envelope images are *not sensed/sense/sensitive/sensor* by human eye [6]. (More than 16.7 million i.e. 224 different colors are *product/produced/production/reproduce* by RGB color model. But human eye can discriminate only a few of them.). This technique is *known/knowledge/knowledgeable/know* as invisible digital watermarking as human eye can not identify the change in the envelope image and the enveloped (Produced after LSB replacement) image [7].

In the decryption *processing/procession/process/processed* k number of embedded envelope images are taken and LSB are retrieved from each of them followed by OR operation to generated the original image.

In this paper Section 2 describes the Overall process of Operation, Section 3 describes the process of k-n secret sharing Visual Cryptography scheme on the image, Section 4 describes the enveloping process using invisible digital watermarking, Section 5 describes decryption process, Section 6 describes the experimental result, and Section 7 draws the conclusion.

**Exercise 3.33.** What is the weakness of the Introduction?

**Exercise 3.34.** Find the definitions for the following words in the Abstract and Introduction.

*Watermarking, Visual cryptography, Pixel*

**Exercise 3.35.** Match the words and definitions.

*discriminate      insecure                      sufficient*

*encryption                      sense*

1. \_\_\_\_\_ *the encoding of electronic data so that it can be transmitted without interception.*
2. \_\_\_\_\_ *enough to meet a need or purpose; adequate.*
3. \_\_\_\_\_ *recognize a distinction; differentiate.*
4. \_\_\_\_\_ *not sufficiently protected; easily broken into.*
5. \_\_\_\_\_ *perceive by a sense or senses.*

**Exercise 3.36.** Translate the highlighted paragraph into Russian.

Grammar focus: Participles I, II

<b>Participle I</b>	active in meaning	are used 1. in relative clauses 2. as the adverbial modifier
<b>Participle II</b>	passive in meaning	

**Exercise 3.37.** What do you think the highlighted words mean?

1. In this current work we have proposed Visual Cryptographic Scheme for color images where the divided shares are enveloped in other images using invisible digital watermarking.
2. Pixel is the smallest unit constructing a digital image.
3. Two transparent objects stacked together, produce transparent object.
4. To add more security to this scheme we have proposed a technique called digital enveloping.
5. Considered as a multiarmed bandit problem, our model corresponds to a specific restless bandit problem that could be used to model different real applications as soon as the efficiency of a given action decreases according to successive frequent uses.

6. Using this technique, the **divided** shares **produced** by k-n secret sharing visual cryptography are **embedded** into the envelope images by LSB replacement.

7. In the decryption process k number of embedded envelope images are **taken** and LSB are retrieved from each of them **followed** by OR operation to generate the original image.

8. A design pattern can be seen as a set of classes, **related** through inheritances, aggregations and delegations, which represents a partial solution to a common non-trivial design problem.

9. Our simulation model is used to compare the different performances of operator selection policies and clearly identify their ability to adapt to such specific operators behaviours and can be used as a surrogate operator model when **designing** new adaptive search algorithms.

10. The experimental study provides interesting results on the respective behaviours of operator selection policies when **faced** to such non stationary search scenarios.

**Exercise 3.38.** Translate the phrases from Russian into English.

1. Насколько автору известно, на данный момент не проводятся исследования...

2. На сегодняшний день в литературе описано только 2 метода...

3. Уже разработано много аналитических методов...

4. Всестороннее рассмотрение всех существующих подходов выходит за рамки этой статьи.

5. Многие ученые уже занимались исследованием этой проблемы ...

6. На самом деле многие авторы исследовали ... и наблюдали ...

7. В литературе нет единого мнения о...

8. Основная цель – установить зависимость...

9. Цель заключается в том, чтобы показать ...

10. Несмотря на недостатки, эти два метода считаются стандартными ...

## ***Revision***

What do you need to know when writing an Introduction?

### ***Self-study activity***

1. Watch the videos: How to write the Introduction: Part 1

[https://www.youtube.com/watch?v=\\_vx1OHRw6To](https://www.youtube.com/watch?v=_vx1OHRw6To)

How to write the Introduction: Part 2

<https://www.youtube.com/watch?v=qRkO--eaYt4>

2. Answer the questions:

Why is the Introduction the first main part of the research paper?

What is the content of the Introduction?

What is the writing style of the Introduction?

How many paragraphs should there be on one page?

What tenses are mostly used?

Write the Introduction for the research paper of your own.

**The phrases that may help you:**

Xs are among the most widely used / commonly discussed / well-known / well-documented

/ widespread / commonly investigated types of ...

X is recognized as being / believed to be / widely considered to be the most important ...

It is well known / generally accepted / common knowledge that X is ...

Recent developments in / findings regarding X have led to ...

X has become a central / an important / a critical issue in...

Previous work has only focused on / been limited to / failed to address ...

there is still a need for / has been little discussion on ...

In this report / paper / review / study we ...



This paper outlines / proposes / describes / presents a new approach to ...

The acronym PC stands for / denotes ...

The subjects (henceforth named / hereafter 'X') are...

The subject, which we shall call / refer to as 'X', is...

This paper is organized as follows / divided into five sections.

The first section / Section 1 gives a brief overview of ...

The second section examines / analyses ...

In the third section a case study is presented / analyzed ...

A new methodology is described / outlined in the fourth section

...

We / I propose a new procedure in Section 4.

Some / Our conclusions are drawn in the final section.

## 1.4. THE STRUCTURE OF THE MAIN BODY OF THE PAPER

### *Focus on theory*

The main body of the paper presents procedures and theoretical basis of the research. It usually consists of several sections and contains the detailed description of the research process. The main body can be structured in the following way:

- Detailed description of the problem under study and its analysis;
- Related work (optional);
- Solution to the problem;
- Results (the findings: positive and negative if there are some).

Taking into account a large amount of information given in this part of the paper, independent work of students is implied. The Master students already know the field of their research and the subject matter of their Master thesis has been chosen. Consequently, they have got a lot of literature sources to read on their major.

**Exercise 4.1** Take any paper on your major and give a detailed description of its main body. You should describe:

- The problem under study;
- Solution to the problem;
- Results.

**Exercise 4.2** Summarize the main body of the paper using the appropriate link words.

### *Self-study activity*

Watch these videos:

1. Good Body Paragraphs

<https://www.youtube.com/watch?v=81lXucfCwpl>

2. How To Write Main Body of Research Paper

<https://www.youtube.com/watch?v=JzoBi8ij0D8&t=54s>

Answer the questions:

1. Why is it necessary to move from general to specific while writing the main body?
2. What are the elements of a good paragraph?
3. What does the breakdown of the main part imply?

Is it necessary to include Way Forward part?

Write the draft of the main body for the research paper of your own.

Write the draft of the main body of the paper of your own.

The phrases that may help you:

In order to identify / understand / investigate / study / analyze X ...

To see / determine / check / verify / determine whether...

We used a variation of Smith's procedure.

In fact / Specifically, in our procedure we ...

Our technique was loosely / partially / partly / to some extent based on

The system is fitted with ...

The software application / program / package used to analyze the data was ...

The following changes / modifications were made ...

This problem can be outlined / posed in terms of ...

The problem is correlated to...

It may easily / simply verified that ...

An X approach was chosen / selected in order to ...

The design of the X was based on

When / As soon as these steps have been carried out, X ...

To assess X / evaluate X / distinguish between X and Y, Z was used

Further analysis / tests showed that

Some / No evidence of X was found ...

The most striking / remarkable result to emerge from the data is that

...

This is in good agreement with...

Our findings appear to be well supported by ...

## 1.5. THE STRUCTURE OF THE CONCLUSIONS

### *Focus on theory*

This section includes the short description of the major findings. After that, future perspectives and/or application of the work are presented. One should keep in mind that the Conclusion section is not the material given in the Abstract or some other part of the paper. This section is usually short (1-2 paragraphs). The items one might find are as follows:

- A brief description of the most essential finding;
- Improvements that can be done;
- Suggestions for further work;
- Tips for policy modification.

## PART A

**Exercise 5.1.** Read and analyze the Conclusions. Focus on the most essential findings described in the section.

A Cross-Cultural Comparative Study of Users' Perceptions of a Webpage: *With a Focus on the Cognitive Styles of Chinese, Koreans and Americans*

**Ying Dong \* and Kun-Pyo Lee**

### **Conclusion**

Nisbett (2001, 2002, 2003) proposed that the thought patterns of East Asians and Westerners differ *great /greatly* and classified these differences as *holistic/ holistically* and *analytically / analyti /*. *Holistically/holistic* -minded people have a tendency to perceive a scene *global /globally*, in other words, to perceive the context and the

field as a whole. They also tend to focus on the relationships between objects and the field, meaning that they are more field-***dependent/dependently***. On the other hand, ***analytically/analytical*** - minded people have a tendency to perceive an object ***separate/separately*** from the scene and tend to assign objects into categories. ***Analytically/analytical***-minded people are more field- ***independently/independent***. In this study, Chinese, Koreans (both holistic thinkers), and Americans (analytic thinkers) were recruited for the experiment. Findings from the analysis suggest that the Chinese, Korean, and American participants employed different viewing patterns when viewing a webpage. The Chinese and Korean subjects showed more similarities to holistic thought patterns, while the American subjects showed more similarities to analytic thought patterns.

The present findings indicate that ***holistically/holistic***-minded people and ***analytically/analytical***-minded people have ***uniquely/unique*** ways of perceiving a webpage. These characteristics of perception reflect some aspects of Nisbett's (2001, 2002, 2003) proposition about cognition. It is suggested that webpage designers should be aware of the cognitive differences existing among ***holistically/holistically-minded*** people and ***analytically/analytical***-minded people, and that webpage design should be carried out according to the target audience's ***specifically/specific*** cognitive style in order to enhance perception and usage of the webpage.

***Exercise 5.2.*** Look up the synonyms for the following words in the dictionary:

- *to perceive* \_\_\_\_\_
- *to propose* \_\_\_\_\_
- *to assign* \_\_\_\_\_
- *to indicate* \_\_\_\_\_
- *to reflect* \_\_\_\_\_

**Exercise 5.3.** Choose the correct highlighted word (either adjective or adverb)

**Exercise 5.4.** What do *in other words*, *on the other hand*, *while* mean in this text?

**Exercise 5.5.** What does the verb *should* mean in the last sentence?

**Exercise 5.6.** Rewrite the sentences using *be likely*. Remember that *be likely* is the same as *to tend* and *to have a tendency*.

1. They also tend to focus on the relationships between objects and the field. \_\_\_\_\_

2. Holistically-minded people have a tendency to perceive a scene globally. \_\_\_\_\_

3. Analytically - minded people have a tendency to perceive an object separately from the scene. \_\_\_\_\_

4. Analytically - minded people tend to assign objects into categories. \_\_\_\_\_

## **PART B**

**Exercise 5.7.** Analyze the content of the Conclusions section and compare the information given in the Abstract with that of the Conclusions.

**Detecting Defects in Object Oriented Designs:  
Using Reading Techniques to Increase Software Quality**

Guilherme H. Travassos<sup>1</sup>, Forrest Shull Michael Fredericks Victor  
R. Basiliw

Experimental Software Engineering Group and Institute for  
Advanced Computer Studies

University of Maryland at College Park

A. V. Williams Building, Bldg#115

College Park , MD, 20742

Fax 301 405 3691

{travassos, fshull, fred, basili}@cs.umd.edu

oCOPPE/UFRJ

C.P. 68511 - Ilha do Fundão

Rio de Janeiro – RJ – 21945-180

Brazil

wFraunhofer Center --Maryland

3115 AgLife Science Surge Bldg

College Park, MD 20742

**ABSTRACT**

Inspections can be used to identify defects in software artifacts. In this way, inspection methods help to improve software quality, especially when used early in software development. Inspections of software design may be especially crucial since design defects (problems of correctness and completeness with respect to the requirements, internal consistency, or other quality attributes) can directly affect the quality of, and effort required for, the implementation.

We have created a set of “reading techniques” (so called because they help a reviewer to “read” a design artifact for the purpose of finding relevant information) that gives specific and practical guidance for



identifying defects in Object-Oriented designs. Each reading technique in the family focuses the reviewer on some aspect of the design, with the goal that an inspection team applying the entire family should achieve a high degree of coverage of the design defects. In this paper, we present an overview of this new set of reading techniques. We discuss how some elements of these techniques are based on empirical results concerning an analogous set of reading techniques that supports defect detection in requirements documents. We present an initial empirical study that was run to assess the feasibility of these new techniques, and discuss the changes made to the latest version of the techniques based on the results of this study.

**Keywords:** Software Engineering Practices, Object Testing and Metrics, Object Oriented Software Quality, Software Inspection.

## 6. CONCLUSIONS

*In this paper/here*, we have described a set of techniques for 1.(review) OO designs. We have applied them in an initial study that has demonstrated their feasibility and provided specific indications for future improvement.

Where possible, we tried to base the OO reading techniques on lessons (learn) from analogous techniques 2.(apply) to requirements documents. Some similarities between the two techniques were by design; *in particular/for example* we adapted a taxonomy of defects that had been useful for requirements reading to focus the design reading on important areas. As a result of the study, *although/however*, we noted other similarities between the two reading techniques. *For example/for instance*, we found that reviewers had difficulty 3.(find) the “right” level of detail for 4.(express) design defects usefully. This result mirrors difficulties that subjects had encountered while 5. (inspect) English-language requirements *or/ but*

which, due to the well-defined notation in which the entire design was expressed, we had not expected to be relevant. Results from this study also may indicate that an effective strategy for 6.(focus) reviewers on semantic aspects of a design is the construction of new artifacts 7.(contain) some of the same information as the document 8.(be reviewed). If this indication is confirmed in further studies it would represent another similarity with the results of requirements reading. **However/no matter**, we did discover crucial differences in design reading that will have to be accounted for in future versions. **As an illustration/ For example**, in requirements reading, syntactic verification is much less important than semantic; the real focus of the inspection is on 9.(verify) the content. When 10.(apply) to OO designs, **however/nevertheless**, syntactic reading becomes much more important, due to the number of separate but inter-related diagrams that must be kept consistent. **Simultaneously/At the same time**, this study has confirmed that syntactic reading can be tedious and should be automated where possible.

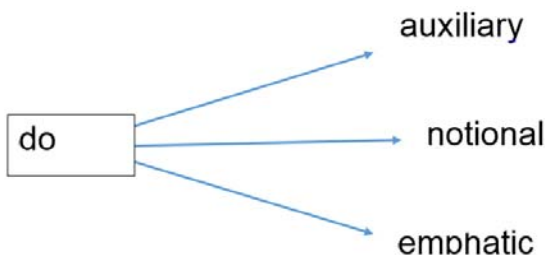
A second important result is concerned with the definition of vertical and horizontal reading. The feasibility study has provided evidence that the use of each type can influence the types of defects 11.(find), and **moreover/in addition**, that both types are necessary to effectively find all defects in the design.

**At present/At this time**, we are working on 12.(incorporate) all of these findings into an improved version of the techniques for further study. We have designed and are putting together a lab package so interested researchers can review the techniques and our other experimental artifacts in more detail. Interested readers can find an initial lab package and the current set of techniques at [http://www.cs.umd.edu/projects/SoftEng/ESEG/manual/tbr\\_package/](http://www.cs.umd.edu/projects/SoftEng/ESEG/manual/tbr_package/)

**Exercise 5.8.** Choose the correct link word, sometimes both of them are possible.

**Exercise 5.9.** Put the verbs in brackets into the correct form.

Grammar focus: the verb **to do**



**Exercise 5.10.** How does the verb **do** differ in these sentences?

1. **Do** the classes exchange messages in the same specified order?
2. However, we *do* present these numbers as subjective indicators of user satisfaction: our subjects seemed to think the techniques were helpful in finding defects.
3. We **do** not present these descriptive statistics as measures of effectiveness.
4. Reviewers in our study **did** have not strong experience in organized development.
5. In order to motivate the students to **do** a good job on the inspection, we gave the students the option of basing their eventual implementation of the system on the design they were inspecting.

## PART C

**Exercise 5.11.** Analyze the content of the Conclusions section and compare the information given in the Abstract with that of the Conclusions.

### Automating GUI Testing for Android Applications

Cuixiong Hu Iulian Neamtii

Department of Computer Science and Engineering

University of California, Riverside, CA, USA

{huc,neamtii}@cs.ucr.edu

#### ABSTRACT

Users increasingly 1. *have relied /rely* on mobile applications for computational needs. Google Android is a popular mobile platform, hence the reliability of Android applications is becoming increasingly important. Many Android correctness issues, however, 2. *fall/is falling* outside the scope of traditional verification techniques, as they are due to the novelty of the platform and its GUI-oriented application construction paradigm. In this paper we 3. *present/will be presenting* an approach for automating the testing process for Android applications, with a focus on GUI bugs. We first 4. *had conducted /conduct* a bug mining study to understand the nature and frequency of bugs affecting Android applications; our study 5. *has been finding finds* that GUI bugs are quite numerous. Next, we 6. *present/were presenting* techniques for detecting GUI bugs by automatic generation of test cases, feeding the application random events, instrumenting the VM, producing log/trace files and analyzing them post-run. We 7. *show/will have shown* how these techniques helped

to re-discover existing bugs and find new bugs, and how they could be used to prevent certain bug categories. We 8. **are believing/ believe** our study and techniques have the potential to help developers increase the quality of Android applications.

## 8. CONCLUSIONS

The number of mobile applications and mobile application users 9. **are growing/had grown** rapidly, which 10. **was creating creates** an impetus for researchers and developers to come up with effective verification techniques 1. \_\_\_\_\_. 2. \_\_\_\_\_ we 11. **perform/will have performed** a bug study to understand the nature and possible remedies for bugs in mobile applications, and construct an automated testing framework for Android applications. Our framework 12. **is combining /combines** automatic event and test case generation 3. \_\_\_\_\_. Our techniques 13. **have proved/was proving** effective for activity, event, and type errors: we 14. **are being able/ have been able** to re-discover existing bugs while finding some new bugs. We 15. **believe/will be believing** our framework can be easily extended 4. \_\_\_\_\_.

**Exercise 5.12.** Choose the appropriate tense form of the verbs.

**Exercise 5.13.** The conclusion lacks in link words. Fill in the link words that might be appropriate.

**Exercise 5.14.** Some parts of the sentences were extracted from the text. Fill in the blanks to make the sentences complete. Remember that one part is not necessary.

- A. Towards this goal,
- B. with runtime monitoring and log file analysis
- C. to ensure the reliability of these applications
- D. to find a broader range of bugs in Android applications

E. to help developers increase the quality of Android applications.

#### **PART D**

**Exercise 5.15.** Analyze the content of the Conclusions section and compare the information given in the Abstract with that of the Conclusions.

#### **Using GUI Ripping for Automated Testing of Android Applications**

Domenico Amalfitano, Anna Rita Fasolino,  
Porfirio Tramontana, and Salvatore De  
Carmine

Dipartimento di Informatica e Sistemistica,  
Università Federico II Napoli, Napoli, Italia  
{domenico.amalfitano,anna.fasolino,  
porfirio.tramontana} @unina.it,  
salvatore.de.carmine@alice.it

Atif M. Memon

Department of Computer Science  
University of Maryland  
College Park, Maryland, USA  
atif@cs.umd.edu

#### **ABSTRACT**

We present *AndroidRipper*, an **automatic /automatically/automated** technique that tests Android apps via their Graphical User Interface (GUI).

*AndroidRipper* is based on a user-interface driven *ripper* that automatically **exploration/explores/explorer** the app's GUI with the aim of 1.(*exercise*)

The **applicant/application/applied** in a structured manner. We **evaluate/evaluation/valuable** *AndroidRipper* on an open-source

Android app. Our results show that our GUI-based test cases are able to detect severe, previously unknown, faults in the underlying code, and the structured exploration **performer/ outperforms/performance**a random approach.

## CONCLUSIONS

We presented *AndroidRipper*, a technique based on GUI 2.(*rip*) for automatic 3.(*test*) of Android applications. Our evaluation using the “WordPress for Android” application revealed four **documentation/ undocumented/document** bugs, automatically detected in less than five hours. This datum shows the **effectiveness/effect/effective** of the technique in 4.(*find*) real bugs and its suitability for 5.(*test*) processes that need to be carried out in a short amount of time. *Moreover*, the experimental data showed that the **proposed/ proposition/proposal** technique is more effective in bug **detect/detective/ detection** than the random testing technique **implemented/implement/implementation** by Monkey.

**Exercise 5.16.** Open the brackets and put the verb in the appropriate form.

**Exercise 5.17.** Chose the correct highlighted word.

**Exercise 5.18.** Put the adjective in brackets into the comparative form, adding any necessary words.

1. Further, from the experimental results of section 5.3, we can see that the average search time of the hierarchical state model (19.4 seconds) is much (little) ... that of the traditional flat state machine (37.1 seconds).

2. To evaluate if the hierarchical GUI model has a (good) readability ... the corresponding model constructed using a flat finite state machine, the average time required to search a GUI state from both models are evaluated.

3. The results revealed that a long events sequence performed (well) ... a shorter events sequence.

4. With over 85% shares of the global market, Android is one of the most popular mobile operating systems in (many) ... 2 billion active devices monthly around the world.

5. A model-based exploration technique produces (good) code coverage ... random testing due to a smaller number of redundant inputs.

## **PART E**

### **DPVK - An Eclipse Plug-in to Detect**

#### **Design Patterns in Eiffel Systems**

Wei Wang, Vassilios Tzerpos<sup>1,2</sup>

*Department of Computer Science*

*York University*

*Toronto, Canada*

#### **Abstract**

Design patterns are not only beneficial to the forward engineering process but also help in design recovery and program understanding, typical reverse engineering activities. In this paper, we introduce DPVK, a reverse engineering tool to detect pattern instances in Eiffel systems. In order to get better detection results, we analyze many different patterns and examine Eiffel software interms of both static structure and dynamic behaviour. DPVK is implemented as an Eclipse plug-in to ensure better compatibility and extensibility.



**Keywords:** BON, Design Pattern, Eclipse, Eiffel, Grok, Reverse Engineering

## Conclusion

**In this paper**, we introduce a diagram catalogue of design patterns. Each pattern diagram includes both a static structure diagram and a dynamic behaviour diagram. The detailed process of generating these diagrams is **also** discussed.

We **also** propose a reverse engineering tool called DPVK, which is used to detect/verify patterns in Eiffel systems. DPVK operates in four stages. The first three stages are static fact extraction, candidate instance discovery and false positive elimination. **Finally**, the fourth stage lets developers manually evaluate and justify the output of previous stages.

**As mentioned in the previous section** the implementation of DPVK is still ongoing. We need to implement the final stage of DPVK and integrate it into the existing architecture. To evaluate DPVK's effectiveness and efficiency, we are going to test it upon all Eiffel implementations in [11]. Also, the EiffelBase library in EiffelStudio is an ideal test base.

DPVK can be enhanced in user interaction and visual presentation.

**For example**, DPVK could automatically generate BON diagrams for each pattern candidate and enable users to work directly upon the diagram to change the candidate's composition.

**As stated earlier**, an Eiffel Development Tool for Eclipse is under development in York University. Once it is ready, DPVK can use it to replace EiffelStudio and EC which are currently used to extract static and dynamic facts about the software system under examination. **In the long run**, DPVK will be extended to detect design patterns in other popular object-oriented language such as Java, C++ etc.

**Exercise 5.19.** Change the highlighted words to their synonyms.

**Exercise 5.20.** Make up 5 questions to the text.

**Exercise 5.21.** Rewrite the sentences using *not only ... but (also)*, *both ... and*, *either ... or*, *as... as*, *rather than*.

1. Each pattern diagram includes a static structure diagram in addition to a dynamic behavior diagram. \_\_\_\_\_

*both ... and* \_\_\_\_\_

2. Design patterns are beneficial to the forward engineering process and also help in design recovery and program understanding, typical reverse engineering activities. *not only ... but* \_\_\_\_\_

3. High-level designs do not attempt to represent details of either the eventual implementation of the system or the “computational world” in which the system exists.

*either ... or* \_\_\_\_\_

4. Although most commonly applied to code documents, software inspections have also been used at earlier stages of the software lifecycle (e.g. in the requirements phase [1,14]) to detect potential problems as early as possible.

*as... as* \_\_\_\_\_

5. Such an approach allows developers to understand the problem instead of trying to solve it.

*rather than* \_\_\_\_\_

**Exercise 5.22.** Translate the sentences from Russian into English using the Gerund:

1. Нужно подтвердить аутентичность и целостность кода перед тем как защитить доступ к защищенным ресурсам.

2. Следующие концепции помогают понять .NETархитектуру.

3. CLR может выполнять как управляемый так и неуправляемый код.

4. Выполнение неуправляемого кода на CLR игнорируют политику безопасности.

5. Эти сервисы могут использоваться большинством приложений, такими как печать и почтовые отправления.

6. Вдобавок к удовлетворению этих основных требований, предлагаются следующие сервисы.

7. Объектно-ориентированное моделирование – это, прежде всего, метод для моделирования и запроса информационной системы на концептуальном уровне.

8. Хотя уже предложено много различных расширений ORM для моделирования процессов и событий, основная задача ORM – моделирование данных.

9. ORM упрощает процесс понимания универсального дискурса (UoD) через использование естественных языков, а также интуитивных диаграмм, которые можно заполнить примерами, и через выражение информации с точки зрения элементарных взаимоотношений.

10. Например, ORM изображает мир с точки зрения *объектов* (сущности или величины), которые играют роли, например, вы сейчас играете роль чтения, эта статья играет роль читаемой.

## ***Revision***

What do you need to know when writing the Conclusions?

### *Self-study activity*

1. Watch the video: How to write the Conclusions  
<https://www.youtube.com/watch?v=rgyeAeE3DGI>
2. Answer the questions:
  - What information does the Conclusions contain?
  - Are there any suggestions for improvements?
  - Are there any recommendations for the future work?
  - Write the Conclusions for the research paper of your own.

#### **The phrases that may help you:**

In conclusion / In summary / In sum / To sum up, our work ...

The evidence from this study suggests / implies / points towards the idea / intimates that ... The results / findings of this study indicate / support the idea / suggest that ...

Our study provides the framework for a new way to do X.

The strength /significance / importance of our work/ contribution lies in ...

X provides a powerful tool / methodology for ...

Our work clearly has some limitations. Nevertheless / Despite this we believe our work could be the basis / a framework / a starting point / a springboard for

This study is the first step / has gone some way towards enhancing our understanding of ...

These observations have several / three main / many implications for research into ...

Research into solving this problem is already underway / in progress ...

Further work needs to be done / carried out / performed to establish whether

## 1.6. ACKNOWLEDGEMENTS

### *Focus on theory*

It is common practice to conclude the paper by giving thanks to the people who were helpful and supportive in doing the research. There are some clichés that are usually used in writing Acknowledgements:

- This work was carried out within the framework of a project
- This work was partly sponsored by...
- The work was possible from a grant from ...
- Support was given by ...
- We would like to thank ...
- We are indebted grateful to ...
- The authors wish to thank Prof. X who gave us much valuable advice ...

Here are some *examples* of Acknowledgements.

1. This material is based upon work supported by the U.S. National Science Foundation under grants CCF-1017204 and CCF-1319695, and by a Google Faculty Research Award.

2. This work is supported by National Key Basic Research Program of China (No. 2016YFB0100900), Beijing Science and Technology Planning Project (No. Z191100007419001), National Natural Science Foundation of China (No. 61773231), and National Science Foundation (CAREER No. 1149783).

3. In acknowledging the various kinds of help and support I have received in the process of this research, I would first of all like to mention with deep gratitude Professor Kun-Pyo Lee, who carefully supervised my work from the beginning of this research. Very special

thanks also to my colleague KiTae Oh, who spent time in developing analysis software for my experiment. I am also very grateful to KAIST (Korea Advanced Institute of Science and Technology), which has provided me with financial support for studying and conducting research. Thanks also to the editor, reviewers and copyeditor from the International Journal of Design for their comments and suggestions on revising this paper. I appreciate all of the above efforts that have gone into seeing that this paper could be published successfully.

4. We would like to thank Corina Pasareanu from NASA Ames Research Center for assisting us with making changes to the Symbolic PathFinder, as well as Aravind Machiry and Mayur Naik from Georgia Tech for assisting us with the setup of Dynodroid. This work was supported in part by awards D11AP00282 from the US Defense Advanced Research Projects Agency, H98230-14-C-0140 from the US National Security Agency, HSHQDC-14-C-B0040 from the US Department of Homeland Security, and CCF-1550206 from the US National Science Foundation.

5. This work was supported by the National Council for Scientific and Technological Development (CNPq)/Brazil (Process 437029/2018-2). PM was supported by CNPq/Brazil (Process 311239/2017-0). WA was supported by CNPq/Brazil (Process 315057/2018-1).

6. This research was supported by the Costa Rican Ministry of Science, Technology and Telecommunications (MICITT). Our thanks to the Empirical Software Engineering (ESE) Group at University of Costa Rica.

7. We thank the authors of the tools we studied for making their tools available and helping with the tools setup. This work was supported in part by NSF grants CCF-1161821 and CCF-1320783 and

by funding from Google, IBM Research, and Microsoft Research to Georgia Tech.

8. This work was partly supported by the National Key Research and Development Program (2017YFB1001904) and the National Natural Science Foundation of China (61772042).

9. The research reported in this paper was supported by the ESPRIT Project 20288 Cooperation Research in Information Technology (CRIT-2): \Evolutionary Real-time Optimization System for Ecological Power Control

***Exercise 6.1*** Express your acknowledgments to your scientific adviser for his or her help in doing the research project.

***Exercise 6.2*** Thank your colleagues for feedback during writing the paper.

***Exercise 6.3*** Give the information about the financial support of your research (funds, grants, etc.)

***Self-study activity***

1. Watch the video: How to Write ACKNOWLEDGEMENTS  
<https://www.youtube.com/watch?v=mwfxfIH2Aw&t=44s>
2. Write the Acknowledgements section to give thanks to the people who were helpful and supportive in doing your work.

## **Part II. SUBMITTING A PAPER TO A CONFERENCE**

### **2.1. CONFERENCE TITLES**

Focus on theory

Before submitting a paper to the conference it is important to read the following sections.

- Conference announcements and call for papers
- Important dates and paper submission
- Additional information: Venue, Accommodation, Tourist information

Scientific papers where you present the results of your research can be published either in the proceedings of the conferences or in peer-reviewed journals. The requirements for these publications (the format, the number of words, font, etc.) are given in the template that can be found on the website of the conference or journal. These requirements and rules must be strictly obeyed otherwise your paper might not be published.

Before publishing a paper in a good peer-reviewed journal it is recommended, though not obligatory, to present new findings at the conference.

The information of most international conferences is published on a website where the organizers make an announcement and provide the authors with the information they might need. First of all, it is important to choose an appropriate conference scanning the titles that can be found on the Internet.

Software engineering Conferences Topics are taken from the Website: <http://www.sostware engineering-conferences.com>

***Here is a list of several general domains in Computer Science.***

***Exercise 1.1.*** Scan the domains in Computer Science given below and select three of them which might be interesting to you. Give your reasons.



## Multidisciplinary or General Events in Informatics

- Training in Computer Science, Courses
- Events for IT Students and Early Career Researchers
- Information Theory, Foundations of Computer Science
- Software Engineering
- Algorithms and Data Structures
- High Performance Computing, Supercomputing, Parallel

## Computing

- Distributed, Edge and Cloud Computing
- Information & Knowledge Management, Big Data

## Computing

- Database Technology and Design, Search Engines and

## Data Mining

- Software Agents and Intelligent Agents in Artificial

## Intelligence

- Soft Computing, Evolutionary Computation, Metaheuristic

## Optimization, Fuzzy Logic

- Information Security and Information Privacy
- Biometric Technologies
- Digital Forensics
- Computing Hardware
- Embedded Systems & Ubiquitous Computing, RFIDs
- Internet of Things (IoT) and Sensor Networks
- Networks, Internet
- Mobile Computing & Telecommunication
- Application Software
- Scientific Computing and Data Science
- Human-Computer Interaction, User Experience
- Natural Language Processing, Computational Linguistics
- Multimedia, Computer Graphics and Visualization, Games
- Social Networks and Social Aspects of Information

## Technology

- e-Business, e-Government, Software as a service (SaaS)
  - e-Learning, Computers in Education
-

- Engineering
- Earth Sciences
- Life Sciences
- Social Sciences
- Healthcare/Medicine
- Didactics, History and Philosophy of Science and Technology
- Continuing Professional Education, Courses

---

**Exercise 1.2.** Scan the announcement of the real conference from the Internet.

Answer the questions.

- Which section would you like to choose to present the results of your current research? Explain your choice.

- Which of the sections would you like to attend? Why?

- Which of the topics do you think are the most popular nowadays?

The 10th International Conference in Software Engineering Research and Innovation (CONISOFT 22)

Universidad Tecnológica Bilingüe Internacional y Sustentable de Puebla \_\_\_\_\_

---

## **JIISIC'2022**

### **Welcome to CONISOFT 2022**

Regarding the COVID 19 pandemic, to safeguard the health of participants, CONISOFT 2022 conference will be hybrid.

The 10th International Conference in Software Engineering Research and Innovation (CONISOFT 2022) will be held at Universidad Tecnológica Bilingüe Internacional y Sustentable de Puebla, October 24-28, 2022. The purpose of the conference is to bring together practitioners and researchers from academy, industry, and government in order to advance the state of the art in Software Engineering, as well as generating synergy between academy and industry, defining a collaboration strategy to support and promote the development of the software industry in Mexico and worldwide. Papers that combine theory and practice are especially welcome.

### **JIISIC**

The Ibero-American Conference on Software Engineering and Knowledge Engineering (JIISIC) is a space for the exchange of knowledge, experiences and good practices on frontier issues related to Computer Science.



From the first version in 2001 until today, JIISIC has summoned more participants every year and has become a meeting of high value for scientists, teachers, technicians and students from all the countries of the region. Its participants form and expand a solid network of contacts, broaden the perspectives of scientific and professional cooperation and define new lines of technology transfer.

# Topics

## Core Themes

- Requirements engineering
- Systems modelling: languages, mechanisms of formalization
- Architecture and software design
- Models and quality systems
- Process models
- Software metrics

## Empirical Software Engineering

- Empirical Methods
- Statistical Methods and Measurement
- Simulation Methods
- Missing Data in Software Engineering
- Experimentation in Software Engineering
- Human Factors in Software Engineering
- 

## Trends in Software Engineering

- Construction of complex systems: recommendation systems
- Software Engineering for very small mobile software applications and context, and very large software ecosystems
- Modelling and knowledge management
- Support methodologies: agents-based methodologies, natural language processing, machine learning
- Software Engineering for Cloud Computing, Big Data, Social Computing

### Interaction with Other Disciplines

- User-centered design: human-computer interaction, usability engineering

- Social Issues in software engineering: psychology, sociology

### Formal Methods

- Theoretical basis for software engineering
- Formal methods for software engineering
- Experimental evaluation and validation
- The split between industry practice and academic research

### Software Engineering Education (SEE)

- Models and Techniques for SEE
- Teaching approaches for SEE
- Tools for SEE
- Strategies and instruments for assessment SEE

### Industry and government experiences

- Cases in industry and government
- Study cases in industry and government

### **Call for papers**

#### Participation Modalities (Types of works)

- Track 1: Scientific papers (original research, basic and experimental).

- Track 2: Work in progress. Papers describing works that are at an early stage of the investigation.

Paper length: Works of all the categories will be submitted as full papers, from 6 pages and up to 10 pages. Papers will be presented in English or Spanish.

Papers must be original works, not have been published before, nor subjected simultaneously to other events.

Submitted papers will be evaluated by an international reviewers committee composed of renowned researchers and professionals, which can be consulted on the Web site of the conference.

### **Submission**

Send your papers using the following link to EasyChair



To write your paper, please use the official template:

### **Final Version**

For sending final version, please follow the directions from IEEE Conference Publishing Services (CPS). CPS WILL BE CONTACTING THE AUTHORS OF THE ACCEPTED PAPERS SOON.

## **Committees**

General Chair

Co-chair

CONISOFT Standing Committee

CONISOFT Steering committee

CONISOFT Reviewers Committee

## **Venue**

CONISOFT 2022 is hosted by Universidad Tecnológica Bilingüe Internacional y Sustentable de Puebla

Boulevard Audi sur S/N, Ciudad Modelo, San José Chiapa, Puebla, México. 75010

## **Code of Conduct**

CONISOFT prohibits retaliation against any member here at CONISOFT who reports or participates in an investigation of a possible violation of our Code, policies, or the law. If you believe you are being retaliated against, please contact our Ethics & Compliance representative.

## ***Here is a list of other conferences from the Internet.***

1. CSASE – CSASE <https://csase.uod.ac>

2022 International Conference on Computer Science and Software Engineering (CSASE). University of Duhok Duhok – Kurdistan Region – Iraq

2. ICSIM 2022 | Yokohama, Japan [http](http://www.icsim2022.org)

International Conference on Software Engineering and Information Management.

3. ICSOFT 2022

Software Engineering and Systems Development, Software Systems and Applications, Foundational and Trigger Technologies.

4. ICSSE 2022  
ICSSE 2022  
2022 Fifth International Conference on Software and Services  
Engineering

**Exercise 1.3.** Look at the titles of the conferences given above and answer the following questions.

- What are the conferences about?
- What kind of specialists might be interested in them?
- Would you be able to present the results of your scientific work at these conferences? Which one?

## **2.2. CALL FOR PAPERS**

When choosing a conference it is important to read Call for papers, where one can find the information about the place where the conference will be held, the dates of the conference, submission deadline and requirements, etc. Here are the fragments of the Calls for paper from different websites where the topics of the sections are covered.

**Exercise 2.1.** Scan the Calls for papers given below in parts A-C and answer the following question.

- What information is given in the Call for papers?

### **A. Call for Research Papers**

• Researchers are kindly invited to submit their papers to the **Second Edition of the International Conference on Computer Science and Software Engineering (CSASE)**. This conference is technically sponsored by IEEE Computer Society. The conference



will be held from **15<sup>th</sup>–17<sup>th</sup> March 2022** at The University of Duhok. The conference program will include paper presentations along with prominent keynote speakers and workshops.

- The accepted papers will be submitted to **IEEE Xplore**.

#### **Notifications and Deadlines**

- **Conference Announcement: 15<sup>th</sup> September 2021**
- **Deadline of Full Papers Submission: 15<sup>th</sup> January 2022**
- **Notification of Reviewing Results: 20<sup>th</sup> January 2022**
- **Submission of Revised Papers: 30<sup>th</sup> January 2022**
- **Final Acceptance Notification: 05<sup>th</sup> February 2022**
- **Deadline for Camera-ready: 15<sup>th</sup> February 2022**
- **Authors' Registration Deadline: 10<sup>th</sup> March 2022**
- **Conference Dates: 15<sup>th</sup> – 17<sup>th</sup> March 2022**
- **Registration Payment will Include:**

1. Welcome Reception.
2. Admission to Scientific Sessions, Workshops, Exhibition

Area.

3. Conference Bag and/or Conference Accessories.
4. Certificate of Attendance.
5. Coffee Breaks.

• All Conference Papers **MUST** have at Least One Registered Author to Present the Paper.

• Please note that the Paid Registration Fee can not be Refunded if the Author is unable to Participate in the Conference for any Reason.

• For all Accepted Papers, the Registration Fee Needs to be Paid. Please note that your Paper **WILL NOT** be Published if the Fee is not Received in Due Time.

• **Methods of Payment (Coming Soon).**

**Submission Guide**

1. Online Submission: The authors are requested to submit their papers via **EDAS**.

2. The submitted paper **MUST NOT** be previously published, and not considered elsewhere for publication.

3. The submitted paper will be double blind peer review.

4. The possible decision on a paper are: **Accepted, Revision (Minor or Major) or Rejected**.

5. All papers will be checked for plagiarism and the similarity check **MUST NOT** exceeding **20%**.

6. Acceptance/Rejection of papers will be decided by the referees of the corresponding conference.

7. The submission is electronically and the size of **A4 with (.docx)** format will only be considered. The paper should be **NOT EXCEEDING 6 PAGES**.

8. IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper **NOT BEING PUBLISHED**.

9. IEEE template and IEEE Copyright Form can be downloaded from **(IEEE-Template)** .

10. Prepare your presentation based on the conference presentation template which can be downloaded from **(Paper ID\_Presenter Name)**.

**Publication Ethics:**

**We firmly believe that ethical conduct is the most essential virtual of any academic. Hence any act of plagiarism is a totally unacceptable academic misconduct and cannot be tolerated.**

### **Paper Submission**

• **Please Refer and Click on One of the Tracks Submission Buttons to Direct You to the Submission Page on EDAS.**

**Submit a Paper to Track 1: Intelligent and Information Systems**

**Submit a Paper to Track 2: Computer Graphics and Visualization**

**Submit a Paper to Track 3: Computer Networks and Communication**

**Submit a Paper to Track 4: Computational Methods and Applications**

### **B. Call for Papers**

• The International Conference on Compiler Construction (CC) is interested in work on processing programs in the most general sense: analyzing, transforming or executing input that describes how a system operates, including traditional compiler construction as a special case.

• Original contributions are solicited on the topics of interest which include, but are not limited to:

- Compilation and interpretation techniques, including program representation, analysis, and transformation; code generation, optimization, and synthesis; the verification thereof

- Run-time techniques, including memory management, virtual machines, and dynamic and just-in-time compilation

- Programming tools, including refactoring editors, checkers, verifiers, compilers, debuggers, and profilers

- Techniques, ranging from programming languages to micro-architectural support, for specific domains such as secure, parallel, distributed, embedded or mobile environments
- Design and implementation of novel language constructs, programming models, and domain-specific languages
- CC is an ACM SIGPLAN conference, and implements guidelines and procedures recommended by SIGPLAN. Prospective authors should be aware of ACM's Copyright policies. Proceedings will be made available online in the ACM digital library from one week before to one week after the conference.

### Submission Guidelines

- **Submission URL:** <https://cc22.hotcrp.com>
- **All** submissions must be made electronically through the conference submission website and include an abstract (100–400 words), author contact information, the full list of authors and their affiliations. Full paper submissions must be in PDF formatted printable on both A4 and US letter size paper.
- All papers must be prepared in ACM Conference Format using the 2-column `acmart` format: use the SIGPLAN proceedings template `acmart-sigplanproc-template.tex` for Latex, and `interim-layout.docx` for Word. You may also want to consult the official ACM information on the Master Article Template and related tools. *Important note: The Word template (interim-layout.docx) on the ACM website uses 9pt font; you need to increase it to 10pt.*
- Papers should contain a maximum of 10 pages of text (in a typeface no smaller than 10 point) or figures, NOT INCLUDING references. There is no page limit for references and they must include the name of all authors (do not use *et al.*).

- Appendices are not allowed, but the authors may submit anonymous supplementary material, such as proofs, source code, or data sets; all supplementary material must be in PDF or ZIP format. Looking at supplementary material is at the discretion of the reviewers.
- Papers may be resubmitted to the submission site multiple times up until the deadline, but the last version submitted before the deadline will be the version reviewed. Papers that exceed the length requirement, that deviate from the expected format, or that are submitted late will be rejected.
- CC follows ACM’s Copyright Policies. Prospective authors should adhere to SIGPLAN’s Republication Policy and to ACM’s Policy and Procedures on Plagiarism.

### **Double-Blind Reviewing Process**

- CC uses a double-blind reviewing process. Authors will need to identify any potential conflicts of interest with PC, as defined in the SIGPLAN policy.
- To facilitate the double-blind reviewing process, submissions (including supplementary material) should not reveal the identity of the authors in any way. Authors should leave out author names and affiliations from the body of their submission. They should also ensure that any references to authors’ own related work should be in the third person (e.g., not “We build on our previous work ...” but rather “We build on the work of ...”).
- The purpose of this process is to help the PC and external reviewers come to an initial judgment about the paper without bias, not to make it impossible for them to discover the authors if they were to try. Nothing should be done in the name of anonymity that weakens the submission or makes the job of reviewing the paper more difficult. In

particular, important background references should not be omitted or anonymized.

### **Artifact Evaluation**

- Authors are encouraged to submit their artifacts for the Artifact Evaluation (AE). The Artifact Evaluation process begins after the acceptance notification, and is run by a separate committee whose task is to assess how the artifacts support the work described in the papers.
- To ease the organization of the AE committee, we kindly ask authors to indicate at the time they submit the paper, whether they are interested in submitting an artifact.
- Papers that go through the Artifact Evaluation process successfully will receive a seal of approval printed on the papers themselves.
- Authors of accepted papers are encouraged, but not required, to make these materials publicly available upon publication of the proceedings, by including them as “source materials” in the ACM Digital Library.
- Additional information is available on the **AE web page**.

### **Publication Date**

- The official publication date is the date the proceedings are made available in the ACM Digital Library. This date may be up to two weeks prior to the first day of your conference. The official publication date affects the deadline for any patent filings related to published work. For those rare conferences whose proceedings are published in the ACM Digital Library after the conference is over, the official publication date remains the first day of the conference.

### **Information to Authors**

- Authors of accepted submissions will be required to choose one of the following options:

- Author retains copyright of the work and grants ACM a non-exclusive permission-to-publish license (and, optionally, licenses the work with a Creative Commons license)
- Author retains copyright of the work and grants ACM an exclusive permission-to-publish license
- Author transfers copyright of the work to ACM
- For more information, please refer to ACM's Copyright Policy and the ACM Author Rights.
- Authors with questions on this Call for Papers are encouraged to contact the Program Chairs by email.

### **C. Call for Papers**

#### **Submissions**

- We invite original submissions in several categories:
  - **Technical design papers** (15 pages incl. references) describe the design of new artifacts, i.e., novel solutions for requirements-related problems or significant improvements of existing solutions.
  - **Scientific evaluation papers** (15 pages incl. references) investigate existing real-world problems, evaluate existing real-world implemented artifacts, or validate newly designed artifacts, e.g., by means such as case studies, experiments, simulation, surveys, systematic literature reviews, mapping studies, or action research.
  - **Vision papers** (6 pages incl. references) state where research in the field should be heading.
  - **Research previews** (6 pages incl. references) describe well-defined research ideas at an early stage of investigation which may not be fully developed.

#### **Submission, Reviewing, and Publication**

- Contributions must be submitted on <https://easychair.org/my/conference?conf=refsq2022>.
- Each submission in the scope of REFSQ will undergo a single-blind review process that will involve at least three members of the program committee.
- The REFSQ 2022 proceedings will be published in Springer's LNCS series. We are also planning a special section with a journal for the best papers.

### **Formatting**

- All submissions must be formatted according to the Springer LNCS/LNBIP conference proceedings template (for LaTeX and Word): <https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines>. As per the guidelines, please remember to include keywords after your abstract.
- Furthermore, to facilitate accurate bidding and a better understanding of the papers, each paper submitted to REFSQ 2022 is *required to have a structured abstract*. The imposed structure demands each abstract have exactly 4 paragraphs with the following content:
  - *Context and motivation*: Situate and motivate your research.
  - *Question/problem*: Formulate the specific question/problem addressed by the paper.
  - *Principal ideas/results*: Summarize the ideas and results described in your paper. State, where appropriate, your research approach and methodology.
  - *Contribution*: State the main contribution of your paper. What's the value you add (to theory, to practice, or to whatever you think that the paper adds value). Also, state the limitations of your results.
- Three examples of structured abstracts are given here.



In the Call for Papers you may also find the Important dates:

- Conference Dates
- Abstract Submission
- Notification of acceptance
- Authors' Registration
- Final Paper Submission

These dates can sometimes be extended.

**Exercise 2.2.** Answer the question.

- Can you specify the phrases that are typically used in the titles of the conferences?

***Self-study activity***

- Find a conference in Software Engineering on the Internet that would be interesting to you and say a few words about it.
- Explain why this conference is a good fit for your specific interests and area of research.

## **2.3. ABSTRACT AND PAPER SUBMISSION**

**Exercise 3.1.** Scan the submission guidelines given below in Parts A and B and answer the following questions.

- What types of papers are mentioned in the guidelines?
- What submission instructions are given in each case?

## PART A

Prospective authors are invited to submit original papers (not being considered for publication elsewhere) in standard format (double column, single-spaced, 10-pt font) describing new theoretical and/or experimental research. Submissions are recommended to have no more than 10 pages (extra pages are subject to surcharge), including figures, tables, and references. Submissions will be judged on originality, significance, interest, clarity, relevance, correctness, and presentation.

### Note for Authors

All submitted papers will be sent to reviewers for a blind review.

The reviewers use the following in evaluating research papers:

- 1 Novel Contribution
- 2 Originality in Thought
- 3 Inferences
- 4 Key Strengths
- 5 Key Weaknesses
- 6 Areas of Improvement
- 7 Presentation/Organization of Research

### Guidelines For Panel Proposal Submission

Panels entail presentation of prepared papers (distributed in advance to a discussant) on a specific topic or theme, followed by structured discussion of those papers at CCECP 2016.

Panels should be comprised by no more than three-four paper presentations, and at least one, but no more than two, Discussants and each paper should have a full abstract and author information.

Panels are guaranteed a 60-80 minute slot (individual paper presentations are expected to run 12-15 minutes).

Each panel requires a chair, which can be self-nominated (during the submission process). They need to prepare a 400-word rationale for the panel proposal and a 75-word panel description for the conference program.

Full Panel Proposal (in word or PDF format) should contain the following:

- Panel Title
- Panel Abstract (200 words)
- Chair
- Discussant
- Presentation Titles
- Papers included in the panel

You may submit your full panel proposal in word/PDF format through EasyChair. Kindly choose panel as category for your proposal submission.

## Registration

Your registration includes:

- Concurrent/Sessions of Technical Program
- Conference Proceedings (for Authors only)
- Welcome Breakfast, Morning and Afternoon Refreshment

Breaks, and Lunch

- Paper Presentation
- Attendance to all sessions
- Portfolio Folder
- GSTF T-Shirt

- Certificate of Presentation
- Complimentary City Tour

## PART B

### Poster & Proceeding Guidelines

Poster size is A0 size (Portrait) . Please refer below for the size dimension.

### Poster Presentation

The following guidelines may prove helpful in the preparation of your poster.

- Use a large font size, and bullet your major points.
- Keep the text to a minimum – most posters contain far too much text.
- Attractive charts, tables and graphics will greatly increase the effectiveness of any poster. Illustrations and tables should be kept relatively simple to maximize legibility. Avoid “artsy” style and keep captions brief.
- Lines in graphs should be heavy. Choose colors that are easily distinguishable from one another. Symbols, letters and numbers should be large enough to be seen from a distance of six feet.
- It’s easier to read a poster if the information is arranged in vertical columns rather than horizontal strips.
- If you wish to show movies as part of your poster presentation, there are now a variety of commercially-available hangers for iPads and similar devices which can be mounted to your poster. Some hangers are like fancy plate holders, others are clear “sleeves”. Bring your own hanger and be sure that it is mounted securely.
- Push pins or Velcro will be provided.

## Poster Evaluation

Students and presenters are required to be at their poster booth 10-15 minutes prior to lunch break and tea breaks. Judges will have a walkabout at random order for evaluation. You are also encouraged to be at your poster booth during the tea breaks to promote and present your work to audiences.

The evaluation on the poster will be on the poster organization, simplicity and concise, straight story line in a logical order and easy to read from appropriate distance. Poster content, figures and results and as well as poster design and layout will be evaluated as well.

## Registration and Payment

Early Bird

Normal

Delegate / Corporate

Oral & Poster Presenter

Accompanying Person/Non-presenting Student

Dinner Ticket

Excursion Visits

## Abstract review criteria

- Relevance to the conference topics
- Significance of the scientific question(s) and results
- Style

- Organization (e.g., the abstract has a clear beginning, middle and end)
- Clear question or hypothesis
- Sufficient background
- The experimental approach and rationale for the approach are clear
- The results are clearly presented
- The interpretation and conclusions are reasonable and logical

On the website of the conferences one can also find additional information about the venue, accommodation, sightseeing of the place, etc.

**Exercise 3.2.** Find the information about other conference presentations on the Internet and make up a list of instructions for each type of presentations.

### **Self-study activity**

1. Watch the video: Call for paper conference  
<https://www.youtube.com/watch?v=qR-Cub4mcqU&t=29s>
2. Study the conference announcements given above and make a glossary you might need when preparing a Call for papers.

### **Real-life practice**

1. Find the information about the conference which is a good fit for your specific interests and area of research.
2. Explore the site of the conference. Give reasons for your choice.  
Which sessions declared in the programme would you like to attend?
3. Fill in the Application form for participation in the conference.  
Write the abstract of your paper in English.
4. Write the abstract of the paper you are going to submit for publication in an international journal.

## APPENDIX 1: Latin Abbreviations in Academic Writing

Abbreviation	Meaning	Example use	Notes for APA Style
Used inside of parentheses only			
cf.	“compare” or “consult” (used to provide contrasting or opposing information)	Abbott (2010) found supportive results in her memory experiment, unlike those of previous work (cf. Zeller & Williams, 2007). She expands on the working memory literature (see also Evans & Potter, 2005).	Never put a comma after. Do not put a period between the c and the f. Use “cf.” to contrast; to compare like things, use “see” or “see also.”
e.g.,	“for example,” (abbreviation for <i>exempli gratia</i> )	Some studies (e.g., Jenkins & Morgan, 2010; Macmillan, 2009) have supported this conclusion. Others—for example, Chang (2004)—disagreed.	Always put a comma after.
etc.	“and so on” or “and so forth” (abbreviation for <i>et cetera</i> )	Students ranked their school subjects (chemistry, math, etc.) in order of preference, first, second, third, and so on, until they had ranked the entire list. A majority ranked science-related subjects (biology etc.) as their second favorite.	Put a comma before if used to end a list of at least two other items, as shown in the example.
i.e.,	“that is,” (abbreviation for <i>id est</i> ; used to give specific clarification)	The experimenters manipulated the order of presentation (i.e., first, second, or third) of the three images as well their size, that is, whether they were small or large.	Always put a comma after.
viz.,	“namely,”	We first replicated our earlier study (viz., Black & Avery, 2008) and then extended it.	Always put a comma after.



vs.	“versus”	The 2 (low vs. high) × 2 (blue vs. green) analysis of variance revealed that the low versus high distinction was not significant.	Exception: With legal citations use <i>v.</i> instead
Used inside and outside of parentheses			
et al.	“and others”	Thomas, Greengrass, and Hopkirk (2010) made several excellent points about goal-seeking behavior. Thomas et al. began with how goals are selected.	Must refer to at least two people because it is a plural phrase
Never used in APA Style			
ibid.	abbreviation for <i>ibidem</i> , used in citations to refer again to the last source previously referenced	_____	Not used in APA Style; instead give each citation using author names as usual.

*Note.* All abbreviations in the first section should be used inside of parentheses only, that is, when you are making a parenthetical statement. Outside of parentheses, spell these expressions out using the definitions given in the Meaning column. The abbreviation “et al.” is used both inside and outside of parentheses. Directions on comma use always apply, whether you are abbreviating or not. Although the abbreviation “ibid.” is not used in APA Style, it is included here because it occurs in non-APA scholarly writing and readers may be otherwise unfamiliar with it. Unless otherwise noted, none of these abbreviations should be italicized.

## APPENDIX 2: The Table of Irregular Verbs

Simple Present	Simple Past	Past Participle
arise	arose	arisen
be	was, were	been
bear	bore	borne
become	became	become
begin	began	begun
bend	bent	bent
break	broke	broken
bring	brought	brought
build	built	built
burn	burned or burnt	burned or burnt
burst	burst	burst
buy	bought	bought
catch	caught	caught
choose	chose	chosen
come	came	come
cost	cost	cost
cut	cut	cut
deal	dealt	dealt
do	did	done
draw	drew	drawn
drink	drank	drunk
drive	drove	driven
eat	ate	eaten
fall	fell	fallen

Simple Present	Simple Past	Past Participle
feed	fed	fed
feel	felt	felt
find	found	found
fit	fit, fitted	fit, fitted
forget	forgot	forgotten
freeze	froze	frozen
get	got	gotten or got
give	gave	given
go	went	gone
grind	ground	ground
grow	grew	grown
hang	hung or hanged	hung or hanged
have	had	had
hear	heard	heard
hide	hid	hidden
hit	hit	hit
hold	held	held
hurt	hurt	hurt
keep	kept	kept
know	knew	known
lay	laid	laid
lead	led	led
leap	leapt or leaped	leapt or leaped
leave	left	
lend	lent	

Simple Present	Simple Past	Past Participle
let	let	left
lie (down)	lay	lent
light	lit or lighted	let
lose	lost	lain
make	made	lit or lighted
mean	meant	lost
meet	met	made
pay	paid	meant
prove	proved	met
put	put	paid
quit	quit	proved or proven
read	read	put
rise	rose	quit
run	ran	read
say	said	risen
see	saw	run
seek	sought	said
sell	sold	seen
send	sent	sought
set	set	sold
shake	shook	sent
shear	sheared	set
show	showed	shaken
shrink	shrank or shrunk	
shut	shut	

Simple Present	Simple Past	Past Participle
speak spend spread take tell think undergo understand win wind withdraw write	spoke spent spread took told thought underwent understood won wound withdrew wrote	sheared or shorn shown or showed shrunk or shrunk shut spoken spent spread taken told thought undergone understood won wound withdrawn written

## APPENDIX 2: Passive voice

- To be + Past Participle (V<sub>3</sub>)

	Simple	Continuous	Perfect
Present	The test is done.	The test is being done.	The test has been done.
Past	The test was done.	The test was being done.	The test had been done.
Future	The test will be done.		The test will have been done.

- **Have something done**

Have + Object + Past Participle (V<sub>3</sub>)

Present Simple	They have it tested.
Present Continuous	They are having it tested.
Past Simple	They had it tested.
Past Continuous	They were having it tested.
Future Simple	They will have it tested.
Future Continuous	They will be having it tested.
Present Perfect	They have had it tested.
Present Perfect Continuous	They have been having it tested.
Past Perfect	They had had it tested.
Past Perfect Continuous	They had been having it tested.

### APPENDIX 3: Non-finite forms of the verb

- **The Infinitive**
- Forms of the Infinitive

	Active	Passive
Present	(to) do	(to) be done
Present Continuous	(to) be doing	–
Perfect	(to) have done	(to) have been done
Perfect Continuous	(to) have been doing	–

- **The Gerund**
- Forms of the Gerund

	Active	Passive
Simple	doing	being done
Perfect	having done	having been done

- **The Participle**
- Forms of the Participle

	Active	Passive
Present Participle Simple	using writing	being used being written
Present Participle Perfect	having used having written	having been used having been written
Past Participle	–	used, written

## APPENDIX 4: Word Formation. Affixation

### Prefixes

Prefixes	Meaning	Examples
<b>a-</b>	not, without	<i>amoral</i>
<b>anti-</b>	against	<i>antisocial</i>
<b>bi-</b>	two	<i>bidirectional</i>
<b>co-</b>	with	<i>co-author</i>
<b>counter-</b>	counter, in opposition to,	<i>counterpart</i>
<b>de-</b>	down; away	<i>decompose</i>
<b>dis-</b>	negation	<i>disagreement</i>
<b>ex-</b>	previous, former	<i>ex-chairman</i>
<b>il- (before l)</b>	not, without	<i>illogical</i>
<b>im- (before b, m, p)</b>	not, without	<i>impractical, imbalance</i>
<b>in-</b>	not, without	<i>insignificant</i>
<b>ir- (before r)</b>	not, without	<i>irrational</i>
<b>inter-</b>	between	<i>international</i>
<b>mal-</b>	wrongly, badly	<i>malformed</i>



Prefixes	Meaning	Examples
<b>mis-</b>	wrongly, badly	<i>misunderstand, misunderstanding</i>
<b>mono-</b>	one	<i>monograph</i>
<b>multi-</b>	many	<i>multidisciplinary</i>
<b>non-</b>	not	<i>non-ability</i>
<b>out-</b>	more, better	<i>outbalance, outgo</i>
<b>over-</b>	excessively	<i>overestimated</i>
<b>post-</b>	after	<i>postdate</i>
<b>pseudo-</b>	false	<i>pseudo-scientific</i>
<b>pre-</b>	before	<i>preanalysis, precalculate</i>
<b>re-</b>	again	<i>recalculate</i>
<b>semi-</b>	half	<i>semi-automatic, semi-natural</i>
<b>sub-</b>	under, less	<i>subdivision</i>
<b>super-</b>	more, big	<i>supercompact, superfast</i>
<b>trans-</b>	into another state or place	<i>transform</i>
<b>un-</b>	opposite	<i>unexpected, unacceptable</i>

Prefixes	Meaning	Examples
<b>under-</b>	not enough	<i>underdeveloped</i>

## Suffixes

### *Forming Nouns*

Suffixes	Meaning	Examples
<b>-age</b>	denoting an action; denoting an aggregate or number of	<i>usage, linkage; percentage</i>
<b>-al</b>	denoting verbal action	<i>refusal</i>
<b>-ance</b>	denoting a quality or state or an instance of one; denoting an action	<i>importance; dissonance; appearance</i>
<b>-ation</b>	denoting an action or an instance of it	<i>exploration</i>
<b>-cy</b>	denoting state or condition	<i>vacancy</i>

Suffixes	Meaning	Examples
<b>-ee</b>	recipient of the action	<i>conferee</i>
<b>-ence</b>	denoting a quality or an instance of it; denoting an action or its result	<i>competence; difference; reference</i>
<b>-er, -or</b>	agent performing the action	<i>publisher, inventor</i>
<b>-ion</b>	denoting verbal action	<i>revision</i>
<b>-ist</b>	denoting a member of a profession or business activity; denoting a person who does something expressed by a verb ending in -ize	<i>plagiarist</i>
<b>-ity</b>	denoting quality or condition	<i>formality</i>

Suffixes	Meaning	Examples
<b>-ment</b>	state, action, the result of	<i>development</i>
<b>-ness</b>	status, condition, quality	<i>effectiveness</i>
<b>-ship</b>	condition, status	<i>membership</i>
<b>-sion</b>	forming nouns of action, condition, etc.	<i>comprehension</i>
<b>-sis</b>	denoting an action or its result	<i>analysis</i>
<b>-tion</b>	forming nouns of action, condition	<i>completion</i>
<b>-ty</b>	denoting quality or condition	<i>loyalty</i>

## ***Forming Adjectives***

Suffixes	Meaning	Examples
<b>-able</b>	able to	<i>acceptable</i>
<b>-al</b>	relating to	<i>logical</i>
<b>-ible</b>	able to be, suitable for being	–
<b>-ic</b>	having the nature of, like	<i>realistic</i>
<b>-ive</b>	tending to; having the nature of	<i>active, conclusive</i>
<b>-ful</b>	full of, characterised by	<i>useful, meaningful</i>
<b>-less</b>	without	<i>useless, meaningless</i>

## ***Forming Verbs***

Suffixes	Meaning	Meaning
<b>-en</b>	denoting the development, creation, or intensification of a state	widen, deepen
<b>-ise</b>	make or become; follow a specified practice	legalise, theorise

## APPENDIX 5: Linking Words and Phrases

Listing/ Sequencing	<p><b>To begin:</b> initially, first, firstly, first of all, in the first place, to start/begin with.</p> <p><b>To continue:</b> secondly, second, thirdly, third, next, then, thereafter.</p> <p><b>To conclude:</b> finally, lastly, eventually, in the end.</p>
Indicating addition	<p>and, both ... and, as well (as), what is more, in addition (to this), additionally, moreover, also, further, furthermore, besides (this), too</p> <p><b>Negative addition:</b> neither ... nor, nor, neither, either.</p>
Highlighting contrast	<p>alternatively, although, but, conversely, despite, even so, even though, for all that, however, in contrast, in spite of, instead,</p> <p>on the contrary, contrary to, nevertheless,</p> <p>nonetheless, notwithstanding, rather, still, on the other hand, otherwise, though, whereas, while</p>
Explaining results/effects	<p>accordingly, as a result/consequence of, consequently, for this reason, hence, so, therefore, thus, inevitably</p>
Giving reasons/causes	<p>a/the consequence of, because, because of,</p> <p>due to, for this reason, as, as a result (of), the effect of ..., the result of ..., since</p>
Making a general statement	<p>generally, in general, as a rule, for the most part, in most cases, normally, on the whole, usually</p>

Stating purposes	<i>so as (not) to, in order (not) to, in case, to, so that, in order that</i>
Referring to other sources:	<i>with reference to, according to, in accordance with</i>
Giving examples	<i>for example, for instance, in particular, such as, thus, a further instance of this is..., an example of this is..., as follows, especially</i>
Rephrasing statements	<i>in other words, more simply, namely, simply, put, to put it differently / another way, such as, that is</i>
Emphasising a point:	<i>in fact, indeed, particularly, especially, interestingly, (un)fortunately, naturally, obviously, unquestionably, clearly, undoubtedly, generally, admittedly, again, to repeat, it should be noted/mentioned (that), more important(ly), most importantly</i>
Making comparisons	<i>similarly, in a similar manner, similar to, likewise, in the same way, too, the same as, equally, correspondingly, in comparison</i>
Providing condition	<i>if, in that case, in case of, in case, unless, provided that, providing that, on condition that</i>
Summarizing	<i>in conclusion, in summary, to sum up, in brief, in short, taking everything into account/consideration, on the whole, all things considered, for the above mentioned reasons.</i>

## MATERIALS USED FOR ANALYSIS

### 1. DOUBLY STOCHASTIC POISSON PROCESS AND THE PRICING OF CATASTROPHE REINSURANCE CONTRACT

Ji-Wook Jang Actuarial Studies, The University of New South Wales Sydney NSW 2052, Australia Tel.: +61 2 9385 3360, Fax: +61 2 9385 1883, E-mail: j.jang@unsw.edu.au

2. A Cross-Cultural Comparative Study of Users' Perceptions of a Webpage: With a Focus on the Cognitive Styles of Chinese, Koreans and Americans  
Ying Dong \* and Kun-Pyo Lee Industrial Design Department, KAIST, Daejeon, Korea

3. Design pattern recovery through visual language parsing and source code analysis

Andrea De Lucia, Vincenzo Deufemia, Carmine Gravino \*, Michele Risi  
Dipartimento di Matematica e Informatica, Universita di Salerno, Via Ponte Don Melillo, 84084 Fisciano (SA), Italy

4. Distributed Object Technology: Security Perspective..  
<https://www.sans.org/white-papers/17/>

5. Visual Cryptography Scheme for Color Image Using Random Number with Enveloping by Digital Watermarking. International Journal of Computer Science Issues 8(3)

Shyamalendu Kandar Indian Institute of Engineering Science and Technology, Shibpur , Arnab Maiti, Bibhas Chandra Dhara Jadavpur University

6. Automating GUI testing for Android applications. Cuixiong Hu, Iulian Neamtii

AST '11: Proceedings of the 6th International Workshop on Automation of Software Test Pages 77–83<https://doi.org/10.1145/1982595.1982612>

7. Object-Role Modeling (ORM/NIAM)

Terry Halpin. Microsoft Corporation, USA

This paper appeared as ch. 4 of Handbook on Architectures of Information Systems, eds P. Bernus,

Details on the book are online at [www.springer.de/cgi-bin/search\\_book.pl?isbn=3-540-64453-9](http://www.springer.de/cgi-bin/search_book.pl?isbn=3-540-64453-9).



8. Automating GUI Testing for Android Applications

Cuixiong Hu Iulian Neamtii. Department of Computer Science and Engineering

University of California, Riverside, CA, USA {huc,neamtii}@cs.ucr.edu

9. Simulating Non Stationary Operators in Search Algorithm

Adrien Goeffon .LERIA, University of Angers (France), Frederic Lardeux LERIA, University of Angers (France), Frederic Saubion LERIA, University of Angers (France)

10. Home > Vol 2, No 2 (2008) > Dong

A Cross-Cultural Comparative Study of Users' Perceptions of a Webpage: With a Focus on the Cognitive Styles of Chinese, Koreans and Americans

Ying Dong \* and Kun-Pyo Lee. Industrial Design Department, KAIST, Daejeon, Korea Home > Vol 2, No 2 (2008) > Dong International Journal of design

11. Detecting Defects in Object Oriented Designs: Using Reading Techniques to Increase Software Quality

Guilherme H. Travassos<sup>1</sup>, Forrest Shull, Michael Fredericks, Victor R. Basiliw

Experimental Software Engineering Group and Institute for Advanced Computer Studies

University of Maryland at College Park A. V. Williams Building, Bldg#115mCollege Park , MD, 20742 Fax 301 405 3691

12. DPVK - An Eclipse Plug-in to Detect Design Patterns in Eiffel Systems  
Wei Wang, Vassilios Tzerpos<sup>1,2</sup> Department of Computer ScienceYork University

Toronto, Canada

## REFERENCES

Adrian Wallwork English for Writing Research Papers. Springer Science+Business Media, LLC 2011

### Internet resources

<https://www.youtube.com/watch?v=jzqCzGXT7hw>

<https://www.sans.org/white-papers/17/>

[www.springer.de/cgi-bin/search\\_book.pl?isbn=3-540-64453-9](http://www.springer.de/cgi-bin/search_book.pl?isbn=3-540-64453-9).

<https://www.youtube.com/watch?v=6-WD4X4IKEs>

<https://www.youtube.com/watch?v=ZVnv6Lv7vjY>

<https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines>.

<https://www.youtube.com/watch?v=N8aBUhGozgo>

<http://www.sostware-engineering-conferences.com>

## **Keys to the exercises**

### **Abstract**

Ex.2.4

Distributed Object Technology: Security Perspective.

Ex.2.6

1.on; 2.at; 3. in ...to; 4. of; 5.to

Ex. 2.7

1C; 2E; 3B; 4D; 5A

Ex. 2.15

1D; 2C; 3E; 4B;5A

Ex.2.18 Automating GUI testing for Android applications

Ex.2.19

1. rely
2. is becoming
3. present
4. conduct
5. finds
6. present
7. show
8. believe

Ex. 2.21

1. application
2. novelty
3. bug

4. prevent

5. increase

Ex. 2.26

1.is; 2.comes; 3.provides; 4. Notes

Ex. 2.23

1.an; 2. more; 3. as; 4. and; 5.is.

Ex. 2.29

1. provides; 2.mapping; 3.model; 4.includes, included; 5.query.

Ex. 2.32

1. in this paper; 2.and; 3.then;4.namely.

Ex.2.33

1. propose; 2. is organized; 3. are identified; 4.are validated; 5.is supported.

Ex.2.34

1.F; 2.C; 3.G; 4.B; 5.D; 6.A; 7.E.

## Introduction

Ex. 3.1

1 Introduction

1.1 ORM: what is it and why use it?

Object-Role Modeling (ORM) is primarily a method for modeling and querying an information system at the conceptual level. In Europe, the method **is** often **called** NIAM (Natural language Information Analysis Method). Since information systems **are** typically **implemented** on a DBMS that is based on some logical data model (e.g. relational, object-relational, hierarchic), ORM **includes** procedures for mapping between conceptual and

logical levels. Although various ORM extensions **have been proposed** for process and event modeling, the focus of ORM is on data modeling, since the data perspective is the most stable and it **provides** a formal foundation on which operations can be defined.

For correctness, clarity and adaptability, information systems **are** best **specified** first at the conceptual level, using concepts and language that people can readily understand. Analysis and design **involves** building a formal model of the application area or universe of discourse (UoD). To do this properly **requires** a good understanding of the UoD and a means of specifying this understanding in a clear, unambiguous way. Object-Role Modeling **simplifies** this process by using natural language, as well as intuitive diagrams that can **be populated** with examples, and by expressing the information in terms of elementary relationships.

ORM **is so-called** because it pictures the world in terms of objects (entities or values) that play roles (parts in relationships). For example, you **are** now **playing** the role of reading, and this paper is playing the role of being read. In contrast to other modeling techniques such as Entity-Relationship (ER) and ObjectOriented (OO) approaches, ORM **makes** no explicit use of attributes. For example, instead of using countryBorn as an attribute of Person, we **use** the relationship type Person was born in Country. This has many important advantages. Firstly, ORM models and queries are more stable (attributes may evolve into entities or relationships). For example, if we decide to later record the population of a country, then our countryBorn attribute needs to be reformulated as a relationship. Secondly, ORM models may be conveniently populated with multiple instances (attributes make this too awkward). Thirdly, ORM is more uniform (e.g. we don't need a separate

notation for applying the same constraint to an attribute rather than a relationship).

ORM is typically more expressive than ER or OO. Its role-based notation **makes** it easy to specify a wide variety of constraints, and its object types reveal the semantic domains that bind a schema together. One benefit of this is that conceptual queries may now be formulated in terms of schema paths, where moving from one role through an object type to another role amounts to a conceptual join (see later). Unlike ORM or ER, popular OO models often **duplicate** information by wrapping facts up into pairs of inverse attributes in different objects. Moreover, OO notations **have** weak support for constraints (e.g. a constraint might have to be duplicated in different objects, or even ignored). Unfortunately, OO models are less stable than even ER models when the UoD evolves. For such reasons, OO models should **be used** only for implementation, not for analysis.

Although the detailed picture provided by ORM is desirable in developing and transforming a model, for summary purposes it is useful to hide or compress the display of much of this detail. Various abstraction mechanisms **exist** for doing this [e.g. CHP96]. If desired, ER and OO diagrams can also be used for providing compact summaries, and **are** best **developed** as views of ORM diagrams. For a simple discussion illustrating the points in this section, see [Hal96].

The rest of this article **provides** a brief history of ORM, **summarizes** the ORM notation, **illustrates** the conceptual design and relational mapping procedures, and **mentions** some recent extensions before concluding.

Ex. 3.4

1.on; 2.to; 3.In ... of; 4.on; 5.of; 6.in;

Ex. 3.5

1.E; 2.C; 3.F; 4.A; 5.B; 6.D

Ex. 3.6

1. technique; 2. implementation; 3. attributes; 4 abstraction; 5. Procedure;  
6. adaptability

Ex. 3.11

1.T; 2.F; 3.F; 4.T; 5.F;

Ex. 3.12

1. Pervasive
2. Impetus
3. Automation
4. Employ
5. Tackle

Ex. 3.13

1.of; 2.to; 3.on; 4.of; 5.with 6.from

Ex. 3.17

1. 2 Selecting the most...
2. 5. Initial parameters...
3. 4. Motivation
4. 1. Contributions
5. 3. Organization of the paper

Ex. 3.18

- a) Precisely
- b) Helpful

- c) Component
- d) Exploit
- e) Explore
- f) Roughly
- g) Comprehensive

Ex.3.19

1.to; 2.on; 3.to; 4.to; 5.in; 6.in; 7.for;8.into

Ex. 3.20

1.C; 2.E; 3.F; 4.G; 5.A; 6.B; 7.D

Ex. 3.24

1. to assess; 2.to separate; 3.to wrap; 4.to encapsulate; 5.to use; 6.be used; 7.to capture; 8.to highlight; 9.be reused; 10.to make; 11.to highlight; 12.to be restructured; 13.to recover; 14.to extract;’ 15.to validate

Ex. 3.26

1. to; 2.in; 3.for; 4.of...from; 5.for...of; 6. - ; 7.with

Ex. 3.27

1. The recovery process is organized in two phases.
2. The results of the case studies are reported and discussed in Section 5.
3. Moreover, the information on the recovered design patterns can improve the system documentation.
4. In this paper we propose an approach for recovering structural design patterns from object-oriented source code.

Ex.3.30

1.d; 2.c; 3.e; 4.a; 5b



### Ex.3.32

#### 1. Introduction

Visual cryptography is a cryptographic technique where visual information (Image, text, etc) gets encrypted in such a way that the decryption can be performed by the human visual system without aid of computers [1].

Like other multimedia components, image is sensed by human. Pixel is the smallest unit constructing a digital image. Each pixel of a 32 bit digital color image are divided into four parts, namely Alpha, Red, Green and Blue; each with 8 bits. Alpha part represents degree of transparency.

Human visual system acts as an OR function. Two transparent objects stacked together, produce transparent object. But changing any of them to non-transparent, final objects will be seen non-transparent. In k-n secret sharing visual cryptography scheme an image is divided into n number of shares such that minimum k number of shares is sufficient to reconstruct the image. The division is done by Random Number generator [4].

This type of visual cryptography technique is insecure as the reconstruction is done by simple OR operation. To add more security to this scheme we have proposed a technique called digital enveloping. This is nothing but an extended invisible digital watermarking technique. Using this technique, the divided shares produced by k-n secret sharing visual cryptography are embedded into the envelope images by LSB replacement [5]. The color change of the envelope images are not sensed by human eye [ 6]. (More than 16.7 million i.e.224 different colors are produced by RGB color model. But human eye can discriminate only a few of them.). This technique is known as invisible digital watermarking as human eye can not identify the change in the envelope image and the enveloped (Produced after LSB replacement) image [7]. In the decryption process k number of embedded envelope

images are taken and LSB are retrieved from each of them followed by OR operation to generated the original image.

In this paper Section 2 describes the Overall process of Operation, Section 3 describes the process of k-n secret sharing Visual Cryptography scheme on the image, Section 4 describes the enveloping process using invisible digital watermarking, Section 5 describes decryption process, Section 6 describes the experimental result, and Section 7 draws the conclusion.

Ex. 3.35

1. encryption
2. sufficient
3. discriminate
4. insecure
5. sense

## Conclusions

Ex. 5.2

1. discern; recognize; understand; notice; comprehend; detect; make out; sense
2. put forward ; present; suggest; advance; come up with; submit;
3. give; set; grant; allocate; give out; consign; allot
4. show; suggest; reveal; display; signal; demonstrate; point to; imply; disclose; manifest; signify; denote
5. show; reveal; express; display; indicate demonstrate; exhibit; manifest

Ex. 5.3

Greatly; holistic; analytic; Holistically-minded; globally; field-dependent; analytically– minded; separately; unique; specific;

Ex. 5.8

In this paper; both; however; both; but; however; for example; however; At the same time; both; both

Ex. 5.9

1.reviewing; 2.applied; 3.finding; 4.expressing; 5.inspecting; 6.focusing; 7.containing; 8.being reviewed; 9.verify; 10.applied; 11.found; 12.incorporating

Ex. 5.12

1.rely; 2.fall; 3.present; 4.conduct; 5.finds; 6.present; 7.show; 8.believe; 9.are growing; 10.create; 11.perform; 12.combine; 13.have proved; 14.have been able; 15.believe

Ex. 5.14

1.c; 2a; 3.b; 4d

Ex. 5.16

1.exercising; 2.ripping; 3.testing; 4.finding; 5.testing

*Издание подготовлено в авторской редакции*

Подписано в печать 02.03.2023 г.

Отпечатано на участке цифровой печати  
Издательства Томского государственного университета  
634050, г. Томск, пр. Ленина, 36, тел. (3822) 529-849.

E-mail: [rio.tsu@mail.ru](mailto:rio.tsu@mail.ru)

Заказ № 5358 от «3» марта 2023 г. Тираж 50 экз.

ISBN 978-5-907572-76-8



9 785907 572768 >