Received: 30 November 2018 | Accepted: 30 January 2019
DOI: 10.1002/smr.2159

RESEARCH ARTICLE - EMPIRICAL

WILEY Journal of Software: Evolution and Process

Integration of properties of virtual reality, artificial neural networks, and artificial intelligence in the automation of software tests: A review

Edgar Serna M.[1] | Eder Acevedo M.[2] | Alexei Serna A.[1]

[1] Instituto Antioqueño de Investigación, Medellín, Antioquia, Colombia
[2] Universidad Cooperativa de Colombia, Medellín, Antioquia, Colombia

Correspondence
Edgar Serna M., Instituto Antioqueño de Investigación, Medellín, Antioquia, Colombia.
Email: eserna@eserna.com

Abstract

The complete automation of software tests has been considered to be an unattainable goal. This article discusses the potential to achieve this goal with recent discoveries and innovations in the areas of virtual reality (VR), artificial neural networks (ANNs), and artificial intelligence (AI). In this study, a theoretical proposal is described to integrate the properties of each of these areas using a process of automation of software tests. This process is based on a classification and description of the properties after consulting the literature, interviews, and dialogs with specialists from Australia, the United States, Germany, and Colombia. In addition to the experiences of the researchers, the construction of two tools is proposed: (1) a robot to design and apply functional tests, and (2) a virtual machine to identify errors in the logical structure of the code. Both tools are expected to replace human factors; the advantage is that the first tool identifies procedural flaws and the second errors of operation.

KEYWORDS

artificial intelligence, artificial neural networks, automation, software testing, virtual reality

1 | INTRODUCTION

For decades, why has one of the objectives of research in computer science been to automate the testing of software? The reality is that our society is software dependent.[1] Due to the current reality of technological development and the increasing consumption of this product, the demand for improvement in software quality has increased. In addition, the rate of software development has increased. This situation has caused industry to rethink its work methodologies to respond to the demands of society and the market, and satisfy the needs of customers and the current demand. However, industry hopes to achieve this goal without investing an excessive amount of time and money.

The scenario becomes increasingly competitive as globalization and the free market offer opportunities to acquire software in any country in the world. Industry must verify and validate each system to evaluate its quality. A set of test cases is designed and manually applied, which involves time-consuming repetitive functions.[2] Automation is used to accelerate processes and reduce costs because the test can provide greater coverage. Another advantage of automation is that it can be executed without supervision with minimal human intervention, at times outside of business hours. In this manner, productivity in the area of quality control is increased, whereas a product can be delivered in less time with cost savings.

Despite decades of research, experimentation, and application, software tests are not fully automated. Although progress has been made and powerful tools are available in the market for this purpose, progress is minimal as society demands quality, reliability, and product safety. Industry attributes this situation to the cost of automation, the lack of qualified personnel, and the ambiguity of the requirements specification documents because they generate delays in development and sufficient time is not available to execute the functional and structural tests established for the system.

Some computational scientists[1,3] are developing innovative ideas that can be integrated into an automation process of tests. They integrate properties of virtual reality (VR), artificial neural networks (ANNs), and artificial intelligence (AI), with research results that demonstrate progress and success stories. One drawback of these investigations is that they do not work in a transdisciplinary manner, that is, they do not experiment, or they experiment almost imperceptibly, and combine the properties of several of these areas. Therefore, the objective of this paper is to analyze, from a theoretical perspective, the potential to integrate, adjust, and harmonize these properties in a process to automate software testing.

The research from which this study originates is developed in two phases: (1) to identify the properties, principles, and representative characteristics of VR, ANN, and AI, and (2) via a broad systematic review of literature, surveys, and dialogs with specialists from Australia, the United States, Germany, and Colombia, and contributions from the knowledge and experience of researchers, to assess their integration in the automation of software tests and describe a theoretical proposal for achieving this integration. The results of the first phase have been published by the following authors: Serna, Serna, and Acevedo[4]; Acevedo, Serna, and Serna[5]; and Serna, Acevedo, and Serna.[6]

2 | METHOD

According to the nature, topic, and scope of the research, the applied method is analytical-inferential but also employs points of view and suggestions from people who are specialized in each of the consulted topics, and the software tests, both manual and automatic. Thus, the research is also cataloged as development. For the collection of data and the analysis of relevance and representativeness of the objective properties, we conducted a systematic review of the literature about the properties of VR, ANNs, and AI. Subsequently, and via joint work between researchers and specialists, the properties were evaluated, complemented, and addressed until the most relevant properties for the achievement of the research objective were identified and documented: the possibility of integrating the properties in the automation of software tests. The search was performed using the ACM, IEEE, Science Direct, and WoS databases, whereas the population of specialists settled with representatives of the industry and academia of Australia, the United States, Germany, and Colombia.

In the sampling, the investigators simultaneously analyzed the findings in the interviews and points of view of the specialists with the collected data, in addition to their experience, until attaining the saturation of theoretical data. Forty-two studies were selected, and 21 specialists in VR, ANNs, AI, software tests, and/or automation of software tests were discussed. During the first two phases (systematic review of the literature and dialogs with specialists) and by permanent integration, the list of properties of each area identified as feasible was built to integrate the properties into the automation of the software tests. Subsequently and after analyzing and discussing the documented properties, the properties that achieved the lowest valuation were eliminated, and the final sample was formed.

Prior to integrating them into the sample, the properties were validated and verified using the advice of specialists by an assessment scale that is defined using the results of the consulted studies, the opinions and recommendations of the specialists, and the experience of the researchers. Consequently, a semantic differential scale of five degrees was defined regarding the probability of integrating each property in the automation: *Very weak*, *Weak*, *Medium*, *Strong*, and *Very strong*. In response to the objective of the research, a triangulation of the assessments was constructed to determine the most likely transdisciplinary integration in the automation of the tests. The results are shown in the next section.

3 | RESULTS

3.1 | Related studies

As mentioned in the introduction, research on the independent correlation among the properties of VR, ANN, and AI in software tests, that is, taking concepts from each individual area and involving them in the phases of the life cycle, is presented. The research that generates this study describes how to interrelate, in a transdisciplinary manner, properties of the three areas in an automation process, which are assumed to be parallel to the life cycle, and the vision of improving the quality, reliability, and safety of the systems. Tables 1–3 present the disciplinary results of representative studies, which are identified in a systematic review of the literature, that integrate properties of VR, ANN, or AI in software tests.

3.2 | Theoretical framework

Given the delimitation of the scope of this investigation and that the objective is to identify interrelations among the properties of VR, NN, and AI, which enable their use in the automation of software tests, conceptual issues such as software engineering, requirements engineering, computational sciences, and formal methods can be considered important, whereas other conceptual issues are not considered. In the opinion of the researchers, the concepts described in the next section are necessary to gain an understanding of the subject related to the research and the content of this article.

TABLE 1 Integration of VR properties in software tests

Ref.	Description
7	They propose the incorporation of characteristics and capabilities of virtual environments in software testing, graphics, support hardware, and interfaces.
8	They describe a technique for performing tests via interactions in VR applications and propose a test architecture.
9	They present a proposal of tests using simulation of virtual reality, in which the user reviews the products in a virtual manner to reduce errors and contribute to a better design.
10	Shows some properties of the VR systems that can be useful in virtual tests, with the idea of overcoming the problems of the manuals.
11	They apply a test proposal based on a VR system and identify a significant improvement in the quality of the product.
12	Investigates the reuse of software in virtual environments and concludes that the development of new tools has increased in the last two decades.
13	They present a tool to use VR in the realization of tests, which can be adapted to test different contexts.
14	They provide new virtual interaction modes, which can be used as tools to automate software testing.

TABLE 2 Integration of ANN properties in software tests

Ref.	Description
15	They propose ANN models to identify the software modules of the system that are most prone to failures to identify modules that need more attention in terms of development.
16	It uses the properties of ANN to predict the effort necessary to execute software tests.
17	Describes an *automated oracle* to support manual testing activities and reduce cost and time.
18	They use the properties of neural networks as an *automated test oracle*.
19	They propose an ANN-based property method to help developers choose between automatic methods and manual methods to run the tests.
20	It presents a new way to use ANNs as *automated agents* to test a system.
21	They describe a model that is based on ANN to identify qualitative and quantitative faults in the modules, as a statistical factor to predict the level at which they can appear in the complete product.
22	They present *oracles* of multiple networks that are based on ANNs, with the objective of automatically executing the tests.
23	Demonstrates how to incorporate ANN properties into tests, giving them meaning via learning techniques with input data.
24	It proposes to take advantage of ANNs in the design of automated test techniques.

TABLE 3 Integration of AI properties in software tests

Ref.	Description
25	They present a description of the common characteristics between AI and software engineering and describe an approach for their automatic integration.
26	Analyze the role of AI in the construction of systems that emulate human behavior, as a basis for the automation of tests.
27	It proposes AI properties as a factor to maximize efficiency and minimize the costs of automated testing.
28	They formulate AI tools as test techniques that can help improve the quality of the software.
29	They analyze the use of AI in software engineering and cover the main relationships between the phases of development and their methods.
30	Explore the relationships between AI and software engineering and establish challenges for the automation of tests.
31	They found that AI for testing not only reduces costs but also guarantees better product quality and more comprehensive testing.
32	They review the techniques developed in AI from the point of view of their incorporation in the design and the automation of test cases.
33	They propose a test system using AI techniques, with the aim of reducing the set of test cases.
34	They focus on the AI techniques in the conceptual stage of the software, which can be associated with the automation of the tests.
35	Work on automation of the tests by taking advantage of AI properties.
36	Analyze the properties of AI and its relationship with improving the quality and reliability of the product by automating the test.
37	They propose various AI approaches to optimize test resources in terms of their automation.

3.2.1 | Software testing

Software is tested to determine if it satisfies the requirements established from the first phase of the life cycle while establishing whether the product in development is the product requested by the client. The analysis consists of answering two questions: (1) Are we correctly building the system? and (2) Are we building the right system? The activities in the first phase are known as Validation and Verification (V&V) activities.[38] In the process, a series of activities is performed to discover and/or evaluate the properties of each element of the system, such as planning, preparation, execution, reporting, and management.[39] Meyers[40] suggest that the goal of the test is to execute a program with the intention of finding errors, whereas Hass[41] affirms that the test comprise a support activity because it does not make sense without the development processes and it produces nothing: *if there is nothing developed, there is nothing to prove*.

The objective of the test is to discover situations that may negatively affect its functioning.[1,2] However, we note that the cost of finding and correcting errors can considerably increase during the life cycle. Therefore, the sooner that errors are discovered, the better their effects—moderate or severe—can be controlled in subsequent stages. The basis of this entire process is the test cases, which are described using attributes to determine their effectiveness and efficiency to find faults. In addition, the attributes must be reusable, that is, they can be easily modified to be executed in scenarios that differ from the original.[42] The error detection process must be economical to perform, analyze, and debug; must be evolutionary; and require a minimum amount of effort. The ability to perform tests is to ensure not only that many defects are identified but also that the tests are well designed to avoid excessive costs and time.[43]

3.2.2 | Automation of tests

Although the literature discusses the automation of tests, its definition focuses on issues such as the creation of a mechanism to automate test cases, that is, the use of special software to execute the tests via a sequence of tests and prepared commands instead of a human tester. With automation, manual activities are assumed to be replaced by their automated equivalents,[44] in such a manner that human participation is not required to execute the test cases. An automated test consists of the use of special software to control the execution of the test cases, and, subsequently, to compare the actual results with the expected results.[45]

Although some organizations have failed in their effort to implement automated tests and have resorted to manual processes. Other organizations have allowed automation to improve the performance of the work team and rapidly increase the quality of systems. Therefore, to obtain the benefits of automation, tests must be carefully selected and applied because the quality of the process is independent of the quality of the tests and the selection of manual or automatic execution does not affect their effectiveness or evolution. Regardless of the intelligence of the planned automation or the ease of execution, if the tests do not achieve anything, then the final results will be evidence that nothing is achieved by the execution. Once implemented, the tests are cheaper because the operating cost is a fraction of the effort required to manually perform the tests. However, the cost of creating and maintaining the test is significant. Thus, intelligently determining when to automate is important because implementation will be cheaper in the long term.[46] In addition, maintenance must be considered because a single update of a set of automated testing cases may have a high cost and may not work in another scenario.

Based on different arguments, companies must decide whether to automate. Although it cannot be classified as the salvation to develop quality systems, automation should be considered. For part of industry, automation is a word that involves greater efficiency, cost reduction, and timely delivery. However, automation is not the objective but the output after application, that is, the system is the final goal. Thus, the task is to decide the role of automation in achieving the final goal. Because automation generates and applies test cases at a lower cost, with better results and fewer reprocesses, it can be considered to be beneficial. In the opposite situation, reprocesses that require additional human testers and a more expensive product are generated, which would have to be considered inappropriate.

3.2.3 | Virtual reality

In a large proportion, users employ VR technology to interact with simulated worlds and environments, in which a sense of illusionary immersion is generated. Other uses of VR are more oriented to engineering, such as mechanics, materials, industrial automation, robotics, architecture, and construction.[47] The concept of VR is not new; the term was already employed by Broderick[48] in his novel. Despite having remained for decades as a prognosis of science fiction, VR could not technologically materialize until recently. Although its usefulness was initially observed, the development of the hardware and the software enabled its massification to become a viable technology.

VR is one of the most prominent cultural tools that humanity has used to capture, transmit and experience their ideas. For many people, VR is the most recent proposal in this timeline. Some of these tools are observed in historical records and range from paintings, narrations, experiences, impressions, radio, and television waves to immersion.[4] Taking advantage of advances in computer Science, VR is a real-time graphic simulation technology that enables users to experience immersion in a *reality that is not their natural environment*. Work in VR has generated useful principles for society and industry, as evidenced in research by Delaney.[49]

Due to these advances, the society of this century begins to migrate many of its activities to digital worlds, which has caused new generations to increasingly abandon the physical interrelations to migrate to the immersion in a network. Part of these activities are mediated by technologies such as telephones, video, messaging, blogs, social networks, games, online universes, forums, and chat channels,[50] which some people refer to as *complex ecology*.[51] One of the reasons for this rapid reception of virtual universes is that the rules, duties, and obligations of the real world disappear in them. In the interior, very different rules must be respected. In most cases, these rules are established by the users, which generates invisible phenomena that are considered to be *cultural discontinuities*.

3.2.4 | Artificial neural networks

ANN comprise a branch of multidisciplinary study in the sciences. Prior to the inception of the first computer, science was interested in studying ANNs to develop artificial models with different applications. For Kohonen,[52] they are networks of simple elements, which are massively interconnected in parallel, usually adaptive and with a hierarchical organization, that attempt to interact with the objects of the real world in the same manner as the interaction of the biological nervous system. These networks behave like a human brain, where information is processed in parallel, with the possibility of learning and generalizing situations that are not included in daily experiences.[53] This process can be considered to be a computational method that is oriented to solve complex problems and has the ability to make predictions in nonlinear relational systems.

The architectural components of ANNs are computational units, whose operation resembles neurons in the brain that are responsible for performing individual calculations and contribute to the total objective of the network, either in learning or training.[5] Due to its architecture, some authors consider it to be a massive information processing system in parallel, which uses distributed control to learn and store knowledge of its environment.[54] This computational ability is achieved due to its design and the ability to use learned knowledge learned to achieve different results that enable them to solve complex problems, extract data, recognize patterns, and approach functions, in contexts different from the context it served to learn.[55]

Because the functioning of ANNs is determined to a great extent by the connections among their elements, they can be trained to execute particular functions by adjustments to the values of these connections. In this manner, they can be applied in computer techniques to derive the meaning of inaccurate data, identify patterns, and detect complex trends. These characteristics have led authors such as Kirkland and Wright[56] to refer to them as *expert systems* because they can be used to make projections based on complex situations, in addition to answering questions, such as *what would happen if?*

To maximize its performance, ANNs must be trained to modify their internal parameters until a function that is as close as possible to what is desired, is implemented. ANNs optimize the set of parameters of its structure via training examples, which teach them underlying rules. During this process, the network incrementally adapts via connections that transport information among its elements and achieves learning via supervision or without supervision. In the first case, a trainer is needed to indicate the answer to be delivered based on the input signal, for which external information is required. In the second case, a trainer or external information is required because it learns using what it locally obtains, which is also known as self-organized learning.[57]

3.2.5 | AI

As a research concept and discipline, AI originated at the Dartmouth conference in 1956 and has attracted the attention of scientists and researchers since this date. For Winston,[58] AI refers to the calculations that enable the technology to perceive, reason, and act; Wachsmuth[3] identifies AI as an area of computational sciences that emphasize these actions. As a research field, AI emphasizes its power to produce intelligent effects or behaviors and seeks progress via systems that perform synthesis prior to analysis. Wachsmuth argues that the goal of AI is to understand natural intelligence by the construction of intelligent machines instead of building intelligent machines based on natural intelligence. For many people, AI involves creating machines that can behave as they do in movies. The main research focus consists of two chapters—scientific and engineering—with clearly demarcated objectives; however, their concepts, methods, and tools are ambiguous.[6]

Despite the initial pessimism about its usefulness and use, AI maintains the promise of developments in the computational sciences to open new paths to science, solve complex problems, and improve the quality of systems. To fulfill this promise, AI must interact with disciplines such as software engineering, within support environments, tools, techniques, and the use of hardware and software.[59] Although some people are skeptical about this type of interaction,[60] the reality is that AI has opened a new field of research in computer science and has achieved high impact with its developments in recent years.

The general image of AI has caused hope and fear for decades and with nuances of imagination, in the ability to speak about a future dominated by intelligent machines. Apart from these beliefs, we must accept that AI techniques will prevail in the IT developments of the coming decades. This prediction is beginning to materialize with initiatives such as *Partnership on AI*, in which Amazon, Google, Facebook, IBM, and Microsoft participate under the slogan *AI for the benefit of people and society*. Subsequent results will be observed from two perspectives: first, the perspective that imitates unique human competences, such as the recognition of letters or the generation of voice, with which most of the current applications are covered; second, the perspective that refers to systems that spontaneously perform creative activities, without supervision or solicitation and that can be characterized as *conscious*. This perspective is not fully developed but is likely to be achieved in a short time.[61]

Due to this perspective, we start to crave intelligent systems that perform analysis and make decisions based on rapidly changing data, that look for patterns, and that handle sensitive tasks that require time and care. We hope that these systems give meaning to the high volume of information that currently circulates, that they generate knowledge, and become a type of *oracle*, which can be consulted for recommendations on *what to do*.[62] This hope is what renders AI extremely and has enabled it to become the engine that presses science and engineering for short-term results. As a science, AI is expected to enable computers to perform tasks, intelligently process information and learn and adapt to changes in each context as a human would.[63]

3.3 | Integration of VR, ANNs, and AI in the automation of tests

In previous work of the researchers, which correspond to the first phase of the investigation of which this article is included, the most significant properties of VR[4] (Table IV), of ANNs[5] (Table V), and of AI[6] (Table VI) were identified. Subsequently and based on the defined assessment scale, the probability of integrating these properties in the automation of the software tests was estimated. Tables 4, 5, and 6 present the average assessment obtained from the results of the review, the opinions of the specialists, and the experiences of the researchers.

TABLE 4 Evaluation of the probability of integrating VR properties in the automation of tests

		Probability of Using Property in the Automation				
Review of the Literature	**Property [4]**	***Very Weak***	***Weak***	***Half***	***Strong***	***Very Strong***
10,64	Virtuality					■
65-68	Modeling and simulation					■
13,69	Interactivity					■
9,11	Immersion				■	
70	Sensoriality		■			
Experience of the authors	Multidimensionality				■	
14	Dynamism				■	
71,72	Multimediality		■			
8	Multiplicity					■
73	Flexibility				■	
74	Immateriality			■		

TABLE 5 Evaluation of the probability of integrating ANN properties in the automation of tests

		Probability of Using Property in the Automation				
Review of the Literature	**Property [5]**	***Very Weak***	***Weak***	***Half***	***Strong***	***Very Strong***
23	Adaptive learning					■
75,76	Self-organization					■
21,77	Fault tolerance				■	
75	Real-time operation					■
17,76	Expansion			■		
75	Convergence			■		
17,20	Validation		■			
Experience of the authors	Complexity			■		
20,21	Nonlinearity		■			
23,24	Training		■			
18,62,78	Memorization					■
79,80	Agility				■	

TABLE 6 Evaluation of the probability of integrating AI properties in the automation of tests

		Probability of Using Property in the Automation				
Review of the Literature	**Property [6]**	***Very Weak***	***Weak***	***Half***	***Strong***	***Very Strong***
31	Interactivity					■
37	Real-time response					■
32,81	Cognitive autonomy					■
82,83	Decision making					■
25,31	Multitask				■	
37,84	Multiple systems				■	
32,81	Autonomous learning					■
36,82	Memorization					■
85	Decision making				■	
Experience of the authors	Rational thinking				■	
84	Creation and execution			■		
35,82	Heuristic search		■			

4 | ANALYSIS AND DISCUSSION

Current software systems are structured by interrelated components that exchange information from and to different dimensions and platforms using different means of communication. This distributed logic in layers has increased the level of complexity of the tests before putting them into operation, which motivates the development teams to include the tests as a parallel process and integrate them into the life cycle from the beginning. As previously discussed, for the software that is currently developed, the application of manual tests is not sufficient, and automation of the process in a high percentage is necessary in such a manner that the functional and structural problems of the system can be identified before the system becomes operational. Thus, the goal of the development team should be to achieve a comprehensive vision of the quality, reliability, and safety of the product before delivering it to a customer and correcting the failures before the user discovers them.

According to the findings of this research, the current tools to automate the tests significantly aids developers and software architects in overcoming these challenges. Although many people address the complex needs of testing, they do not identify and solve most problems or structure and implement an adequate set of test cases. One of the most influential reasons, given the type of systems and the problems they are currently solving, is that these systems have been designed and constructed with a disciplinary approach. In this manner, they may adequately respond in a certain type of test for a specific system but may not respond to another type of test that has greater importance in other systems.

The proposal is to integrate properties from different disciplines, which have demonstrated utility and applicability in tests, to build an automated tool to design, implement, analyze, and update test cases according to the type of test (functional or structural) and type of system to be tested. The test is self-delimited, self-dimensioned, and self-programmed to satisfy the test requests regardless of the system on which it applies. In this article, and according to the results obtained in the investigation, the proposal is structured as follows:

1. Consider the properties of VR, ANNs, and AI, which are valued as strong or very strong, with greater probabilities of integration into a tool of this type, and define them with criteria that enable them to be interrelated in an automated solution. If the properties are repeated in several of these disciplines, only one property is considered to be a reference. In the case of VR, the selected properties are defined as follows[4]:
    - *Immersion*. It refers to a state of the self, in which consciousness is interpenetrated in an absorbing virtual world that represents the simulation of a real world.
    - *Multidimensionality*. The virtual worlds must accurately reflect real dimensions to enable the brain to properly assimilate its physical stay within a nonphysical entity.
    - *Dynamism*. It is the capacity for transformation and adaptation, which have the elements of a virtual world and enables a user to experience a higher level of reality in relation to what he would obtain in a static world.
    - *Flexibility*. It is the faculty of the elements of an environment to easily adapt to the needs of the design, the objective, and the level of immersion that a person wishes to recreate in the virtual world.
    - *Virtuality*. It is the content of a given world that can exist only in the mind of the author or can be shared with others.
    - *Modeling and simulation*. The model represents the system and the simulation of its operation over time. The objective is to visualize the eventual real effects of the alternative conditions in its operation to select courses of action.

- *Interactivity*. It is the degree to which technology creates environments in which users communicate, synchronously or asynchronously, and interact via a medium.
- *Multiplicity*. It is related to presence and space in a virtual system, where the duration of progressions and interrelationships are derived from multiple points via multiple media and mediated by a geometric idea.

The properties of the selected ANNs are described as follows[5]:

- *Fault tolerance*. Capacity of a system to continue functioning, even if some of its components fail.
- *Agility*. It is the ability of a system to effectively change its location and position and maintain total control of its components to adapt to a change in the context of action.
- *Adaptive learning*. Ability to learn from contexts and develop agility to adapt to each context.
- Self-*organization*. It is a capacity that allows a system, after learning the patterns of a domain, to recognize similar patterns, even if a person has not trained for them.
- *Real-time operation*. Ability of a system to recognize patterns in real time and simultaneously update its components because they work in parallel.
- *Memorization*. Although it is a human intellectual activity, in systems it consists of locating and saving patterns, data, and sequences that can be consulted when necessary.

The selected AI properties are described as follows[6]:

- *Multitasking*. It is the property of systems to simultaneously allow or execute several processes while sharing resources, data, patterns, and sequences.
- *Multiple systems*. It is the ability of a system to interrelate the functions of different systems, regardless of their similarity, to achieve a certain objective.
- *Cognitive autonomy*. It refers to the functions, processes, and states that enable systems to understand, infer, make decisions, plan, and learn based on the information they process.
- *Decision-making*. It is a process that systems execute to define, collect and process data, generate information, infer alternatives, and select an alternative based on the most appropriate identification to solve a specific problem.
- *Autonomous learning*. It is the ability of a system to self-direct, regulate, update, and define itself, with the aim of structuring and implementing a solution based on its learning.

2. Integrate the properties in the automation of any of the types of tests (functional or structural) or both types of tests. This integration is determined according to the definitions and recommendations of the authors and the specialists and the experiences of the researchers. For this research, the results are shown in Figure 1. The criteria for performing this integration are as follows: (1) analysis of the results of the selected studies in the research, considering that they were separately obtained using only properties of each area but that the context in which it is experimented is the automation of the tests; (2) the recommendations of the specialists, which are defined in the interviews and dialogs, about the properties they accept as useful in an integration process for automation; and (3) the experience of researchers, accumulated in their research on solving the problem of automation in software testing.

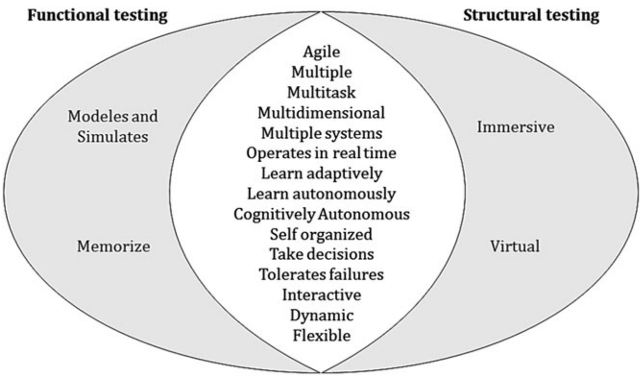

FIGURE 1 Properties of VR, ANN, and AI that can be integrated into the automation of the tests

Most of the evaluated properties are located in the *common zone* for functional and structural tests, which indicates the possibility of integrating them to achieve the objective of the investigation. This feature enables developers to acquire the necessary knowledge to optimize the code and the time and costs necessary for the structuring, design, and development of the tools and enables developers to project the updates and innovations that the end product will need in the course of its useful life. They can assimilate each property as a requirement in the life cycle, and therefore, when the properties are modified, the requirements will also be modified.

3. Structure the tool that can be developed to satisfy these *requirements*, considering the objectives and characteristics of functional and structural tests. The following alternatives are recommended:
   - For structural tests, whose objective is to run a system to evaluate its operation, agility, safety, and reliability, as well as capabilities such as speed of response, robustness, comprehensiveness, and data handling, and which, by definition, are executed by a person from a set of test cases, the proposal is to build a *robot* that replaces it. This tool is designed based on the previously established requirements, considering the structure, semantics, and grammar of the programming language with which the system is developed. The robot analyzes the information that it collects from each programming context and each domain of the system to construct a set of test cases that must be applied under these circumstances. Then, the robot compares the obtained results with the expected results and presents a report of the test that indicates the identified failures.
   - The objective of the functional tests is to review the code to determine procedural errors according to the internal structure defined by the grammar and semantics of the applied language, in addition to the mathematical logic of the included equations, functions, or formulas. They differ from the functional tests because the functioning of the system is not important; rather, the logic of the code enables the achievement of its objectives. Thus, the recommended tool should be a *virtual machine* that replaces the person in charge of reading the code and who typically is the developer. Taking advantage of the properties located in Figure 1 for structural and common tests, this machine goes through the code analyzing its procedures while interacting with the functional testing robot to determine if the faults, which it identifies in the operation of the system, are attributed to code errors. In this case, the robot virtually shows the error line, continues to interact with it for the duration of the tests, and will do the same when the error is generated by procedures in the code.

This multitasking and interrelational work of the tools allows both to acquire cognitive autonomy at the same time as autonomous and adaptive learning, with which they can make dynamic, flexible, and interactive decisions to solve the faults, while memorizing the response procedures for each context. On the one hand, the robot operates in real time and adapts the set of test cases to each system to be tested, and, on the other hand, the virtual machine self-organizes the most common errors in the code based on the language used. These multiprocesses enrich the cognitive heritage of the tools through the exchange of information that each one converts into knowledge to execute its functions in the most appropriate way in each test situation.

5 | CONCLUSIONS AND FUTURE WORK

The software industry, society, and customers are currently at a crossroads: to guarantee the quality, safety, and reliability of this technological development; however, before implementing and placing it into operation, an adequate test plan is necessary. The method by which tests are structured, designed, and executed consumes time and resources and increases the final cost of development. An alternative is to automate the tests and involve them as a parallel activity as much as possible to prevent it from being reduced to a process at the end of the life cycle. Some authors have experienced different methods for achieving this goal. However, as many of these authors accept it, their work falls short because they do not employ a transdisciplinary approach. They approach the solution from a disciplinary perspective using the properties that best fit the purpose of the work.

In this research, the automation of software testing can be achieved using a transdisciplinary analysis and integration of properties from different areas, whose functionality has been separately tested and validated. The goal of this work is to propose, from a theoretical conception, properties of VR, ANNs, and AI that can be interrelated and integrated in the automation of tests by the design of tools that are oriented to functional and structural tests with transdisciplinary vision.

The challenges generated by this theoretical approach range from the possibility of materializing it in practice with the development of tools to validating and verifying that they work. With discoveries and innovations[86] that have been achieved in VR, ANNs, and AI, researchers are convinced that processes conducive to achievement should be initiated. The descriptions presented in the results of this study are an ideological basis for other initiatives, which provide continuity via community work to materialize the initiatives.

The most important future work that is projected from the results is the conceptualization, design, and development of the robot and the virtual machine. We need to form a broad, transdisciplinary, and non-narrow-minded team that addresses the challenge as an achievable and feasible product to build on what is known and expected. In addition, the authors hope that this work will attract the attention of interested people and that they will be encouraged to form the previously mentioned team.

ORCID

Edgar Serna M. https://orcid.org/0000-0001-5306-4444

REFERENCES

1. Serna E. *Functional Test of Software—A Constant Verification Process*. Fondo Editorial ITM: Medellín; 2013.
2. Serna E, Arango F. Effectiveness analysis of the set of test cases generated with the requirements by contracts technique. In: *V Congreso Colombiano de Computación*. Cartagena, Colombia; 2010.
3. Wachsmuth I. The concept of intelligence in AI. In: Cruse H, Dean J, Ritter H, eds. *Prerational Intelligence—Adaptive Behavior and Intelligent Systems without Symbols and Logic*. The Netherlands: Kluwer Academic Publishers; 2000:43-55.
4. Serna E, Serna A, Acevedo E. Description, analysis and discussion of the properties of virtual reality. *Virt Real* 2018; In press.
5. Acevedo E, Serna A, Serna E. Principles and characteristics of artificial neural networks. In: Serna E, ed. *Desarrollo e Innovación en Ingeniería*. 2a ed. Medellín: Editorial Instituto Antioqueño de Investigación; 2017:173-182.
6. Serna A, Acevedo E, Serna E. Principles of artificial intelligence in computer science. In: Serna E, ed. *Desarrollo e Innovación en Ingeniería*. 2ª ed. Medellín: Editorial Instituto Antioqueño de Investigación; 2017:161-172.
7. Bierbaum A, Just C. Software tools for virtual reality application development. In: *Course Notes for SIGGRAPH 98 Course 14, Applied Virtual Reality*. 1998.
8. Bierbaum A, Hartling P, Cruz C. Automated testing of virtual reality application interfaces. In: 9th workshop on virtual environments; 2003; Zurich, Sswitzerland
9. Dwivedi S, Ikonomov P, Milkova, ED. Beta testing of design product using virtual reality simulation. 2003. Online [Feb 2018].
10. Foit K. Introduction to use virtual reality visualisations in the exploitation and virtual testing of machines. *J Achieve Mater Manufact Eng*. 2007;25(2):57-60.
11. Zhao X, Xu L, Li H. Software testing applications based on a virtual reality system. *J Electron Sci Technol China*. 2007;5(2):120-124.
12. Steed A. Proposals for future virtual environment software platforms. In: Coquillart S, Brunnett G, Welch G, eds. *Virtual Realities*. New York: Springer; 2008.
13. Avgoustinov N, Boller C, Dobmann G, Wolter B. Virtual reality in planning of non-destructive testing solutions. In: Bernard A, ed. *Global Product Development*. Berlin: Springer; 2011:705-710.
14. Falcão C, Soares M. Application of virtual reality technologies in consumer product usability. *Lecture Notes on Computer Science*. 2013;8015:342-351.
15. Khoshgoftaar T, Allen EB, Hudepohl JP, Aud SJ. Application of neural networks to software quality modeling of a very large telecommunications system. *IEEE Trans Neural Netw*. 1997;8(4):902-909.
16. Dawson C. An artificial neural network approach to software testing effort estimation. *Trans Info Comm Technol*. 1998;20:1-11.
17. Vanmali M, Last M, Kandel A. Using a neural network in the software testing process. *Int J Intel Syst*. 2002;17(1):45-62.
18. Mao Y, Boqin F, Li Z, Yao L. Neural networks based automated test oracle for software testing. *Lect Notes Comp Sci*. 2006;4234:498-507.
19. Smilgyte K, Nenortaite J. Artificial neural networks application in software testing selection method. *Lect Notes Comp Sci*. 2011;6678:247-254.
20. John J. A performance based study of software testing using artificial neural network. *Int J Logic Based Intel Syst*. 2011;1(1):45-60.
21. Sandhu P, Lata S, Grewal D. Neural network approach for software defect prediction based on quantitative and qualitative factors. *Int J Comp Theory Eng*. 2012;4(2):298-303.
22. Shahamiri S, Kadir W, Ibrahim S, Hashim S. Artificial neural networks as multi-networks automated test oracle. *Auto Softw Eng*. 2012;19(3):303-334.
23. Yogi A. Reformation with neural network in automated software testing. *Int J Comp Trends Technol*. 2013;4(6):1816-1819.
24. Sathyavathy V. Evaluation of software testing techniques using artificial neural network. *Int J Eng Comp Sci*. 2017;6(3):20617-20620.
25. Rech J, Althoff K. Artificial intelligence and software engineering: status and future trends. *Künstliche Intelligenz*. 2004;18(3):5-11.
26. Zeigler B, Muzy A, Yilmaz L. Artificial intelligence in modeling and simulation. In: Meyers R, ed. *Encyclopedia of Complexity and System Science*. Heidelberg: Springer; 2009:344-368.
27. Staats M. The influence of multiple artifacts on the effectiveness of software testing. In: IEEE/ACM international conference on automated software engineering. 2010 Antwerp, Belgium
28. Larkman D, Mohammadian M, Balachandran B, Jentzsch R. General application of a decision support framework for software testing using artificial intelligence techniques. In: Second KES International Symposium IDT. 2010 Baltimore, USA
29. Meziane F, Vadera S. Artificial intelligence in software engineering—current developments and future prospects. In: Meziane F, Vadera S, eds. *Artificial Intelligence Applications for Improved Software Engineering Development: New Prospects*. New York: Information science reference; 2010:278-299.
30. Harman M. The role of artificial intelligence in software engineering. In: *First International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering*. Zurich, Switzerland: Cognizant; 2012.
31. Rauf A, Alanazi M. Using artificial intelligence to automatically test GUI. In: 9th International Conference on Computer Science & Education. 2014; Vancouver, CANA
32. Shankari K, Thirumalaiselvi R. A survey on using artificial intelligence techniques in the software development process. *Int J Eng Res Appl*. 2014;4(12):24-33.
33. Kire K, Malhotra N. Software testing using intelligent technique. *Int J Comp Appl*. 2014;90(19):22-25.

34. Sharma S, Pandey K. Integrating AI techniques in SDLC. In: *Third International Symposium on Women in Computing and Informatics*. 2015 Kochi, India
35. Pawar P. Application of artificial intelligence in software engineering. *J Comp Eng*. 2016;18(3):46-51.
36. Bhateja N. Various artificial intelligence approaches in field of software testing. *Int J Comp Sci Mobile Comput*. 2016;5(5):278-280.
37. Bhateja N, Sikka S. Achieving quality in automation of software testing using AI based techniques. *Int J Comp Sci Mobile Comput*. 2017;6(5):50-54.
38. Boehm B. Software risk management. *Lect Notes Comp Sci*. 1989;387:1-19.
39. ISO/IEEE. *ISO/IEEE 29119—Part I International Standard. Software and systems engineering/software testing, concepts and definitions*. USA: IEEE; 2013.
40. Myers J. *The Art of Software Testing*. New York: Wiley; 1979.
41. Hass A. *A Guide to Advanced Software Testing*. Boston: Artech House; 2008.
42. Karthikeya S, Rao S. Adopting the right software test maturity assessment model. In: *Cognizant 20-20 Insights*. New Jersey, USA: IEEE; 2014.
43. Anya P, Smith G. Qualitative research methods in software engineering. *Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software (RACCIS)*. 2014;4(2):14-18.
44. Haugset B, Hanssen G. Automated acceptance testing: a literature review and an industrial case study. In: Agile Development Conference; 2008 Toronto, CANA; 27-38.
45. Safronau V, Turlo V. Dealing with challenges of automating test execution. In: Third International Conference on Advances in System Testing and Validation Lifecycle. 2011 Barcelona, Spain; 14-20.
46. Furtado A, Meira S, Gomes M. (2014). Towards a maturity model in software testing automation. In: The Ninth International Conference on Software Engineering Advances; 2014 New York, USA; 282-285.
47. Mujber T, Szecsi T, Hashmi M. Virtual reality applications in manufacturing process simulation. *J Mater Process Technol*. 1834-1838;2004:155-156.
48. Broderick D. *The Judas Mandala*. USA: Pocket Books; 1982.
49. Delaney N. *The market for visual simulation/virtual reality systems*. Mountain View: Cyberedge Information Services; 2004.
50. Hernández A, Pérez K. Criteria for verifying and validating mechanisms in the development of videogames. *Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software (RACCIS)*. 2017;7(1):7-12.
51. Kow Y, Young T. Media technologies and learning in the StarCraft eSport community. In: Conference on Computer Supported Cooperative Work. 2013; San Antonio, USA; 387-398.
52. Kohonen T. An introduction to neural computing. *Neural Netw*. 1988;1(1):3-16.
53. Singhal D, Swarup K. (2011). Electricity price forecasting using artificial neural networks. *Int J Electric Power Energy Syst*. Elsevier 2011; 33(3): 550-555.
54. Haykin S. *Neural Networks. A Comprehensive Foundation*. New York: Prentice Hall; 1999.
55. Kulkarni A. *Artificial Neural Networks for Image Understanding*. New York: John Wiley & Sons; 1993.
56. Kirkland L, Wright R. Using a neural network to solve testing problems. *IEEE Aerospace Electron Syst Magazine*. 1997;12(8):36-40.
57. Maimon O, Last M. *Knowledge Discovery and Data Mining—The Info Fuzzy Network (IFN) Methodology*. Boston: Kluwer Academic Publishers; 2000.
58. Winston P. *Artificial Intelligence*. Massachusetts: Addison-Wesley; 1993.
59. Partridge D. *The relationships of AI to software engineering*. London, UK: IEEE Colloquium on Software Engineering and AI; 1992.
60. Last M, Kandel A, Bunke H. *Artificial Intelligence Methods in Software Testing. Series in Machine Perception & Artificial Intelligence*. USA: World Scientific Publishing; 2004.
61. Mair C, Kadoda G, Lefley M, et al. An investigation of machine learning based prediction systems. *J Syst Softw*. 2000;53(1):23-29.
62. Aggarwal K, Singh Y, Kaur A, Sangwan O. A neural net based approach to test oracle. *ACM Softw Eng Notes*. 2004;29(3):1-6.
63. Gelly S, Kocsis L, Schoenauer M, et al. The grand challenge of computer Go: Monte Carlo tree search and extensions. *Comm ACM*. 2012;55(3):106-113.
64. Alsmadi I. How much automation can be done in testing? In: Alsmadi I, ed. *Advanced Automated Software Testing: Frameworks for Refined Practice*. USA: Information Science Reference; 2012:1-29.
65. Stone B, Pegman G. Robots and virtual reality in the nuclear industry. *Serv Robot*. 1995;1(2):24-27.
66. Stuart R. *The Design of Virtual Environments*. New York: McGraw-Hill; 1996.
67. Gong Y, Wang B, Wang W. The simulation of grinding wheels and ground surface roughness based on virtual reality technology. *J Mater Process Technol*. 2002;129(1-3):123-126.
68. Foit K, Swider J, Mastrowski D. The project of an off-line, remote programming system for Mitsubishi Movemaster industrial robot. In: proceedings of the XIII international scientific and technical conference. 2006; Sevastopol, Ukraine
69. Kuliga S, Thrashb T, Daltonc C, Hölscher C. Virtual reality as an empirical research tool—exploring user experience in a real building and a corresponding virtual model. *Comp, Environm Urban Syst*. 2015;54:363-375.
70. Gausemeier J, Rammig F, Radkowski R, Krupp A, Müller W. Virtual and augmented reality for systematic testing of self-optimizing systems. In: *International Design Conference*; 2010; Dubrovnik, Croatia
71. Godoy J. Have the software testing a future? *Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software (RACCIS)*. 2012;2(1):18-25.
72. Hackett M. The changing landscape of software testing. *LogiGear Magazine*. 2013. Online [Mar 2018].
73. Cigniti. How to amplify the impact of virtual reality with a robust test strategy? 2002. Online [Feb 2018].
74. Banerjee P. Virtual reality and automation. In: Nof S, ed. *Springer Handbook of Automation*. Berlin: Springer; 2009:269-278.

75. John J, Wanjari M. Performance based evaluation of new software testing using artificial neural network. *Int J Sci Res*. 2014;3(5):1519-1534.

76. Shahamiri S, Nasir W. Intelligent and automated software testing methods classification. In: 4th postgraduate annual research seminar. 2008; Skudai, Malaysia

77. Saraph P, Kandel A, Last M. Test set generation and reduction with artificial neural networks. In: Last M, Kandel A, Bunke H, eds. *Artificial Intelligence Methods in Software Testing*. New Jersey: World Scientific; 2004:101-132.

78. Ye M, Feng B, Zhu L, Lin Y. Neural networks based automated test oracle for software testing. In: 13th international conference on Neural information processing. 2006 Hong Kong, China

79. Su Y, Huang C. Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models. *J Syst Softw*. 2007;80(4):606-615.

80. Kumar A. Dynamic test case generation using neural networks. 2005. Online [Mar 2018].

81. Kire K. Software testing using intelligent technique. *Int J Comp Appl*. 2014;90(19):22-25.

82. Moore A, Solomon R, Barney M. Automation and Ai—from man to machine. 2017 Computer Science. Online [Feb 2018].

83. Platz W. *Beyond continuous testing with artificial intelligence*. Vienna: Tricentis; 2017.

84. Nordström J. *The Future Will Be Automated*. Stockholm: 3GAMMA; 2016.

85. Matthews K. How machine learning is impacting the way we test software. 2017 TechZone 360°. Online [Feb 2018].

86. Roldán M, Giraldo J, Betancur A. State of the art and methodological approach to product innovation evaluation in organizations of the telecommunications industry. *Revista Actas de Ingeniería*. 2016;2:210-218.

How to cite this article: Serna M. E, Acevedo M. E, Serna A. A. Integration of properties of virtual reality, artificial neural networks, and artificial intelligence in the automation of software tests: A review. *J Softw Evol Proc*. 2019;e2159. https://doi.org/10.1002/smr.2159