Decision Support Systems xxx (xxxx) xxxx

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

A decision-making support system for Enterprise Architecture Modelling

Ricardo Pérez-Castillo*, Francisco Ruiz, Mario Piattini

Information Technologies and Systems Institute (ITSI), University of Castilla-La Mancha, Paseo de la Universidad 4, 13071 Ciudad Real, Spain

ARTICLE INFO

Keywords:
Enterprise Architecture
Viewpoint
Genetic algorithm
Reverse engineering
ArchiMate

ABSTRACT

Companies are increasingly conscious of the importance of Enterprise Architecture (EA) to represent and manage IT and business in a holistic way. EA modelling has become decisive to achieve models that accurately represents behaviour and assets of companies and lead them to make appropriate business decisions. Although EA representations can be manually modelled by experts, automatic EA modelling methods have been proposed to deal with drawbacks of manual modelling, such as error-proneness, time-consumption, slow and poor re-adaptation, and cost. However, automatic modelling is not effective for the most abstract concepts in EA like strategy or motivational aspects. Thus, companies are demanding hybrid approaches that combines automatic with manual modelling. In this context there are no clear relationships between the input artefacts (and mining techniques) and the target EA viewpoints to be automatically modelled, as well as relationships between the experts' roles and the viewpoints to which they might contribute in manual modelling. Consequently, companies cannot make informed decisions regarding expert assignments in EA modelling projects, nor can they choose appropriate mining techniques and their respective input artefacts. This research proposes a decision support system whose core is a genetic algorithm. The proposal first establishes (based on a previous literature review) the mentioned missing relationships and EA model specifications. Such information is then employed using a genetic algorithm to decide about automatic, manual or hybrid modelling by selecting the most appropriate input artefacts, mining techniques and experts. The genetic algorithm has been optimized so that the system aids EA architects to maximize the accurateness and completeness of EA models while cost (derived from expert assignments and unnecessary automatic generations) are kept under control.

1. Introduction

Enterprise Architecture (EA) is a valuable tool to represent and manage IT and business in a holistic way by establishing connections among technology concerns and business, strategical, and motivational aspects. EA Management (EAM) is "a discipline for proactively and holistically leading enterprise responses to disruptive forces by identifying and analysing the execution of change toward desired business vision and outcomes. EA delivers value by presenting business and IT leaders with signature-ready recommendations for adjusting policies and projects to achieve target business outcomes that capitalize on relevant business disruptions" [1]. One of the major benefits of EAM perceived by companies is that it enables them to achieve the effective alignment between business and IT [2]. EAM provides the viewpoints mechanism, which can be used to holistically understand any system's fundamental organization by means of all embodied viewpoints, such as stakeholders and their concerns, processes, services, applications, IT resources, and so forth [3]. Viewpoints are an abstraction mechanism that represents a set of EA models, each aimed at a particular type of stakeholder and addressing a particular set of concerns (e.g., IT infrastructure for IT architects, or Goals and Motivations for managers).

The business-IT alignment cannot be easily achieved and, when it is, its agile adaptation is not trivial in a world with changing markets and volatile technologies [4]. Companies consequently tend to (re)define business goals and processes, along with the respective functionality of their IT (micro)services, by (re)developing and operating them in a continuous way [5,6].

EA modelling has traditionally been carried out manually by experts. However, manual EA modelling has several inconveniences [7], such as error-proneness, time-consumption, slow and poor re-adaptation and cost. The root causes of these inconveniences are the subjective opinion provided by experts when they create EA views, which might lead to models with missing elements (false negatives) and irrelevant elements (false positives), along with a lack of automation. Because of those flaws, recent studies [5,7,8] state the need to automate EA modelling through the use of different reverse engineering and mining techniques in order to discover EA models from a wide variety of artefacts (e.g., information systems, enterprise service bus, databases, or

* Corresponding author.
E-mail addresses: ricardo.perez@uclm.es (R. Pérez-Castillo), francisco.ruizg@uclm.es (F. Ruiz), mario.piattini@uclm.es (M. Piattini).

https://doi.org/10.1016/j.dss.2020.113249
Received 30 August 2019; Received in revised form 7 January 2020; Accepted 12 January 2020

source code).

Automatic EA modelling techniques deal with subjectivity issues and quick re-adaptation needs. What is more, these techniques are more efficient than manual modelling and are, thus, cheaper than manual techniques. However, these automatic techniques hardly ever work well for the modelling of EA viewpoints with the most abstract concepts, such as motivational or strategy viewpoints [9]. This is because this kind of information is usually in the hands of directors and managers and change over time in volatile environments. Even when this kind of information is available in documents or artefacts, it is not as up to date as the information handled by experts.

As a result, the latest research and industry appear to demand hybrid approaches [7], in which some EA viewpoints can be extracted automatically (specifically those at the process, application and infrastructure levels), and can be combined with manual modelling. This will allow experts to review and refine the EA models discovered and build other missing models for viewpoints on the business, strategical or motivational layers.

For such hybrid approaches, there are a vast number of input artefacts and experts that organizations might potentially use in EA modelling efforts [7]. However, even if organizations understand the importance of having accurate and complete EA models, these organizations have limited resources [10] and are, therefore, reluctant to assign more resources than those strictly required. It is important to stress that assigning several experts to EA manual modelling projects (which may last for weeks or even months) is very expensive. Moreover, reverse engineering and mining tools (for automatic modelling) usually entail even higher costs owing to licensing or associated development projects [9].

With regard to the aforementioned challenge, we also detected the problem that clear relationships are missing among the input artefacts, mining techniques and the EA viewpoints to be extracted, and there are additionally blurry relationships between experts' roles and the viewpoints to which they are able to contribute. This lack of knowledge prevents organizations from making informed decision regarding assignments in EA modelling projects and from choosing appropriate mining techniques and their respective input artefacts, which leads to undesirable situations. First, EA modelling projects exceed the budget because unnecessary stakeholders are involved. Second, EA modelling projects run without the optimal sources of knowledge, as a result of which the outgoing EA models have some flaws. Finally, even if appropriate input artefacts are selected, there is no information about which mining technique should be used to extract as many elements as possible for a model based on a certain viewpoint (without mentioning the lack of information about how to combine such automatic modelling with expert-aided modelling).

This paper presents a piece of research that is part of a long-term investigation developed using the Design Science Research Method (DSRM) [11–13]. The main contribution of this paper is a decision support system, whose core is a genetic algorithm, for computing optimal plans for EA modelling. This algorithm takes as input two lists with the information system artefacts and experts available in the company. As output, it provides a sorted set of steps consisting of automatic modelling using one mining technique on one of the available artefacts, as well as manual modelling by one of the experts selected.

This decision support system is designed and developed through the elicitation and definition of the aforementioned non-explicit and/or unknown relationships: input artefacts and EA elements that could be used in a certain viewpoint, depending on the mining technique that might be employed; as well as experts and those EA elements for which they are responsible and are able to create in models for different viewpoints.

With regard to the first type of relationships, this research considers insights obtained in a previous systematic mapping study conducted in [7], while the second type of relationships are attained through an analysis of the available viewpoints and stakeholders involved according to the ArchiMate specification [14]. ArchiMate is considered, by many authors, as the de facto EA modelling specification. However, the definition of similar relationships could be defined according to one of many other available EA framework/languages [15]. The expression 'viewpoint elements' will hereafter be employed to refer to those ArchiMate elements that can be used in EA models concerning the specific viewpoint. The same ArchiMate element can be used in models with different viewpoints.

The decision support system (the artefact according the DSRM) has been developed as a web application (so-called ArchiRev-VS) that, bearing in mind the aforementioned relationships plus the genetic algorithm, is able to compute the degree of accomplishment of all the EA viewpoints (as described in the ArchiMate standard) regarding input artefacts and experts that are available as well as propose optimized EA modelling plans.

As further contribution, the genetic algorithm as the core of the decision support systems has been validated and optimized by using different configurations. Considering this, ArchiRev-VS aids EA architects as regards which input artefacts to use in automatic EA modelling, and which experts to consider in manual modelling.

The remainder of this paper is organized as follows: Section 2 summarizes related work. Section 3 presents the DSR method followed in this research. Section 4 describes the decision-making support system for EA modelling, both the viewpoint coverage computation, and the optimized EA modelling plan computation. Section 5 describes the optimization of the genetic algorithm included in the ArchiRev-VS tool. Finally, Section 6 provides a discussion regarding the implications of this proposal.

2. Related work

As we introduced, viewpoints represent a set of EA models that abstract information for a particular stakeholder and set of concerns. The viewpoint abstraction is stated in ISO/IEC/IEEE 42010 (see Fig. 1), on which most of the EA frameworks are based. Viewpoints make it possible to manage complexity in EA, since it can be modelled in terms of a set of different viewpoints and the correspondences between them.

This section is organized in four subsections, for both manual and (semi)automatic modelling of viewpoints. Section 2.1 summarizes modelling of viewpoints in a manual way. Related to the manual modelling, Section 2.2 explains how relationships between experts and viewpoints have been investigated to make decisions about modelling. Section 2.3 explains how other works have addressed (semi) automatic modelling of EA viewpoints, while Section 2.4 explains how relationships between input artefacts and viewpoints have been addressed in the literature.

2.1. Manual modelling of viewpoints

Although a certain amount of work regarding EA modelling exists (most of which is performed manually), there is no comparable amount of work addressing EA viewpoints as the central aspect of EA modelling. For example, Steen et al. [16] present the design of a tool environment for viewpoint-oriented EA, which supports the definition, generation, editing and management of architectural views. The main benefit of this tool environment is the possible integration of other domain-specific modelling tools managed at a certain company. Atkinson and Tunjic [17] analyse how the ArchiMate viewpoint framework works as regards the Orthographic Modelling approach, which applies the idea of dimension-based view identification and selection in a pure, comprehensive form. The comparison shows that there are some weaknesses in ArchiMate's viewpoint framework, and the authors explain how it could be adapted. This research is performed using nAOMi tool, which provides an orthographic modelling environment in which to explore viewpoint information.

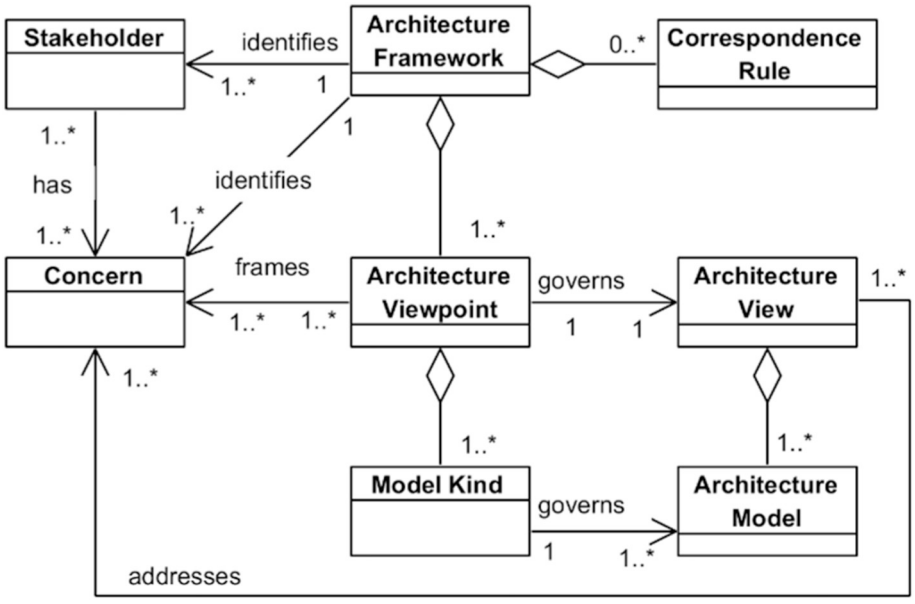


Fig. 1. Architecture viewpoint and related concepts according to ISO/IEC/IEEE 42010.

2.2. Stakeholders and viewpoint modelling decisions

Regarding the implication of stakeholders in EA modelling, Hacks et al. [18] demonstrated that there are homogenous concerns among stakeholders with regard to EA deliverables, although with some differences in the case of the stakeholders' hierarchical level. These authors state the need for a more differentiated understanding of stakeholder concerns as regards EAM. In this respect, Puspitasari [19] provides a stakeholder's expected value analysis scheme with a priority matrix for different EA stakeholder profiles. This approach serves to solve stakeholders' potentially conflicting expected values by prioritizing the value fulfilment based on a stakeholder's contributions and concerns. Mezzanotte and Dehlinger [20] propose a behaviour-driven EA requirements quality management program designed to encourage stakeholder collaboration and participation in EA. These authors suggest using an Architectural Design Plan (ADP) to define (among other things) which stakeholders will be selected and assigned to the EA project team, in addition to a rationale with which to justify why other stakeholders are not assigned to the project. Unfortunately, this proposal focuses on the procedure and does not provide much information about stakeholders' assignments under different circumstances.

2.3. (Semi)automatic modelling of viewpoints

With regard to the (semi)automatic modelling of EA viewpoints, some proposals consider specific reverse engineering or mining techniques, as stated in a systematic mapping study provided by Pérez-Castillo et al. [7] and as also stated by Farwick [8]. Unfortunately, most of these techniques focus on some viewpoints in isolation as output and consider certain artefacts as input. Nevertheless, there is not much guidance as regards how to perform the whole viewpoint map in organizations in an integrated manner, as suggested in this study. One exception to this is the proposal made by Brosius et al. in [21]. These authors suggest guidance through an analysis framework in order to coordinate heterogeneous and potentially conflicting stakeholder concerns, since most EAM initiatives reach only specific stakeholders or selected contexts. The analysis framework considers coordination by means of its underlying formal and informal mechanisms, which are implemented by artefacts, and also through the use of artefact modalities. Although the proposal in [21] is promising, it focuses on the coordination aspect rather than on the decision-making process. What is more, it was not specially developed for (semi)automatic EA modelling.

2.4. Input artefacts and viewpoint modelling decisions

"Almost since the inception of computing, there has been interest in the question of how technology will change management work" [22]. There is some research about how to make decisions about EA viewpoint modelling. Romero and Vallecillo [23] deal with the specifications of correspondences between views regarding different viewpoints. These authors argue that most EA frameworks consider these specification correspondences in a very simplistic way and that they are not totally explicit. This work proposes some well-formed rules with which to complement correspondence specifications in multi-viewpoint modelling approaches. Similarly to our proposal, Ruiz et al. [24] employ a simulation-based optimization method. Actually, that proposal uses two multi-objective evolutionary algorithms, although the main application is to manage changes in the context of IT service management instead of EA modelling.

There are other works that focus on decision-making processes for modelling individual concerns. For example, Zapata et al. [25] use sentiment analysis to make decisions about the information structure viewpoint. Kitsios and Kamariotou [26] provided a literature review of business strategy modelling based on enterprise architecture. However, this work does not provide a specific decision-making process for modelling the strategy viewpoint. Alfonso-Robaina et al. [27] propose a system based on fuzzy decision rules for modelling enterprise architecture and strategic management viewpoint. Similarly, Sohaib et al. [28] a 2-tuple fuzzy linguistic decision-making method to make decisions about IT infrastructure based on cloud computing services. In general, the majority of EA mining approaches in literature assume that viewpoints are populated from artefacts on a one to one basis [7]. However, this is rarely the case in practice. In general, information contained in a viewpoint needs to be obtained from multiple artefacts of different types. Thus, the real problem is not choosing which individual artefact to mine for a given viewpoint but which combinations of artefacts to mine. The same happens for stakeholders and manual EA modelling. In this context, the contribution of our paper is the decision-making process as regards the usage made of EA viewpoint modelling by stakeholders and also based on mining techniques and specific artefacts.

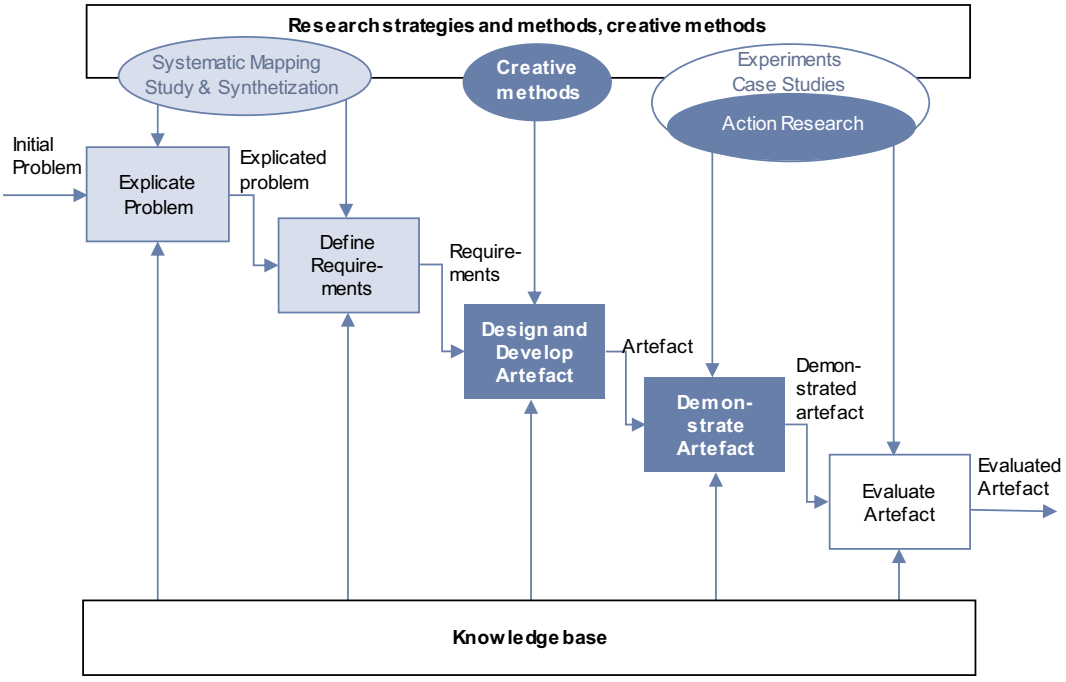


Fig. 2. The design science research method adapted from ref. [11].

3. Research method

As mentioned previously, this paper describes a specific piece of research that is part of a multiyear research project, which is framed in DSRM [11,12,29,30]. Design Science 'is the scientific study and creation of artefacts as they are developed and used by people with the goal of solving practical problems of general interest' [11]. The main goal is the design and investigation of artefacts in context. The artefacts are designed to interact with a problem context in order to improve something in that context [13].

DSRM advocates the use of the workflow shown in Fig. 2. Specifically, this research can, according to the classification provided by [11], be viewed as 'development- and evaluation-focused design science research'.

This DSRM type is instantiated because this research focuses principally on the design and development of an artefact (the third activity in Fig. 2). The main goal is, therefore, to design and develop an artefact using both research and creative methods to establish relationships between input artefacts and EA viewpoint elements, depending on the mining technique that might be employed, and experts and certain elements that are able to model specific EA viewpoints. The two first activities (problem explanation and requirements definition) are not within the scope of this paper and were accomplished by means of a systematic mapping study carried out previously [7]. The results of this literature review allowed us to explain the problem introduced (the first activity in Fig. 2). Moreover, an analysis of the results of the preliminary study in question, along with its synthetization, provided the requirements definition (the second activity in Fig. 2). A better understanding of the requirements can be accomplished through the description of the use cases for the target system (see Section 4.1).

In the third DSRM activity carried out in our research and the scope of this paper, the artefact is designed and developed (cf. Section 4). As we introduced, the target artefact is a decision-making support system (with a genetic algorithm as a core), named ArchiRev-VS, which is able to calculate optimum EA modelling plans. In the fourth DSRM activity, we demonstrate the application of the artefact through the application of the systems under different scenarios with different inputs (cf. Section 5). The end goal of this execution is actually the optimization of the genetic algorithm. This phase is mainly performed following the technical action research [31].

4. ArchiRev-VS. Decision-support system for Enterprise Architecture Modelling

This section explains how ArchiRev-VS has been designed and built following DSRM principles.

4.1. Use cases

ArchiRev-VS might be employed by companies as an EA management system, but also it might be useful as regards training architects how to use different viewpoints in each situation, depending on the types of stakeholders and aspects to focus on. For this reason, we believe ArchiRev-VS can be used as a decision-making support system. In particular, ArchiRev-VS can be used in four uses cases (see Fig. 3), which can be performed in isolation or in combination. Table 1 shows the four scenarios formulated as decision-making problems.

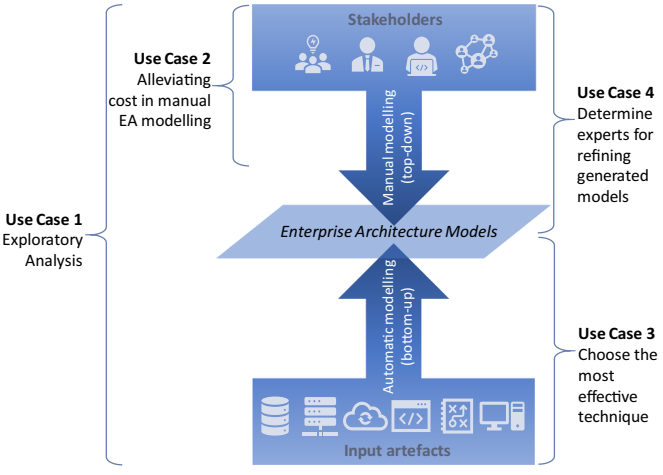


Fig. 3. Summary of ArchiRev-VS usage scenarios.

Table 1
Summary of ArchiRev-VS usage scenarios formulated as decision-making problems.

Scenario	Problem	Cause of the problem	Decisions to be made	How decisions are made
1	Uncertainty of EA modelling.	Absence of previous knowledge/experience in hands of EA architects.	What stakeholders or input artefacts must be considered in the company for EA modelling?	Exploratory analysis without previous knowledge.
2	Excessive cost of EA modelling.	Over assignments of stakeholders for manual modelling/ unnecessary development of modelling tools.	What stakeholders and input artefacts (with the respective mining technique) should be considered?	Detection of unnecessary stakeholders or input artefacts to achieve the same level of viewpoint coverage.
3	Unnecessary application of mining techniques in EA modelling.	Unknown possibilities of mining techniques (regarding what viewpoint elements can be obtained).	What is the minimum, necessary set of mining techniques for modelling the desired EA viewpoints?	Detection of better mining techniques in terms of viewpoint coverage.
4	Unknown experts necessary for manual intervention after automatic EA modelling.	After mining techniques have been applied, it is difficult to be what experts are necessary for modelling uncompleted viewpoints.	What experts should be chosen to complete EA viewpoint as much as possible?	List the minimized set of stakeholders in EA modelling plans.

4.1.1. Use case 1. Exploratory analysis

This system can be used by experts to perform exploratory analyses. This means that enterprise architects can play with different input artefacts and roles so as to attain an overall idea of how different parameters affect the coverage of each viewpoint. This scenario can be performed without knowledge of which input artefacts and roles are available in the company. For example, a company whose IT infrastructure is not service-oriented will probably not have an enterprise service bus available as an input artefact from which to extract EA models. However, this exploratory analysis could provide architects with suggestions concerning the future to-be status of the EA. Migrating to a service-oriented architecture may allow this company to not only model, but also re-adapt their EA models automatically from the enterprise service bus.

4.1.2. Use case 2. Alleviating costs in manual EA modelling

Many companies still address EA modelling manually. Moreover, most of these companies do not consider changing to automatic modelling owing to the associated initial cost. For example, they have to invest money in order to develop ad hoc mining techniques or buy expensive tools. In this scenario, an important use case is to detect over-assignments to EA modelling projects, i.e., more people than is strictly necessary are in charge of modelling EA. Executions of this system with different configuration could detect the minimum number of different roles that are necessary to model each EA viewpoint. The main implication is that the total cost of manual EA modelling projects can be reduced without reducing the accuracy and completion of outgoing models.

4.1.3. Use case 3. Choosing the most effective reverse engineering/mining technique

Companies that already automatically generate some EA models might be interested in ArchiRev-VS in order to compute different scenarios with their current input artefacts to check how different reverse engineering and mining techniques work with each of the artefacts available. The computed viewpoint coverage could, therefore, be used as a decision-making mechanism with which to choose better techniques, i.e. those with which to achieve/discover most of the elements involved in each viewpoint. Additionally, the genetic algorithm directly provides an optimized plan with a list of steps (automatic or manual) to be performed.

4.1.4. Use case 4. Determining experts for the refining of automatic-extracted models

Another scenario in which ArchiRev-VS could be used is in companies that already use some kinds of reverse engineering or mining techniques to discover or readapt EA models. In this case, manual operation after automatic modelling is necessary in most cases, as stated at the beginning of this paper. This system can compute different roles that could be necessary to obtain the missing elements: those cases in which automatic modelling was not able to discover all the viewpoint elements. Through the computation of the optimized plan for EA modelling (by means of the genetic algorithm) experts involved in each modelling or refining step are delivered.

4.2. ArchiRev-VS modules

ArchiRev-VS has been developed as a utility module of a bigger EAM suite (ArchiRev) in order to generate EA models using different artefacts by using reverse engineering techniques. ArchiRev is available online at [32] and ArchiRev-VS is the decision-making support system designed and implemented in this research. Fig. 4 shows the overall architecture of ArchiRev-VS, along with some details of the technology stack. ArchiRev-VS follows a Model/View/Controller (MVC) architecture. The model layer manages the aforementioned relationships among the elements of the viewpoints and the stakeholders and input artefacts that might be employed to model EA (both manually and

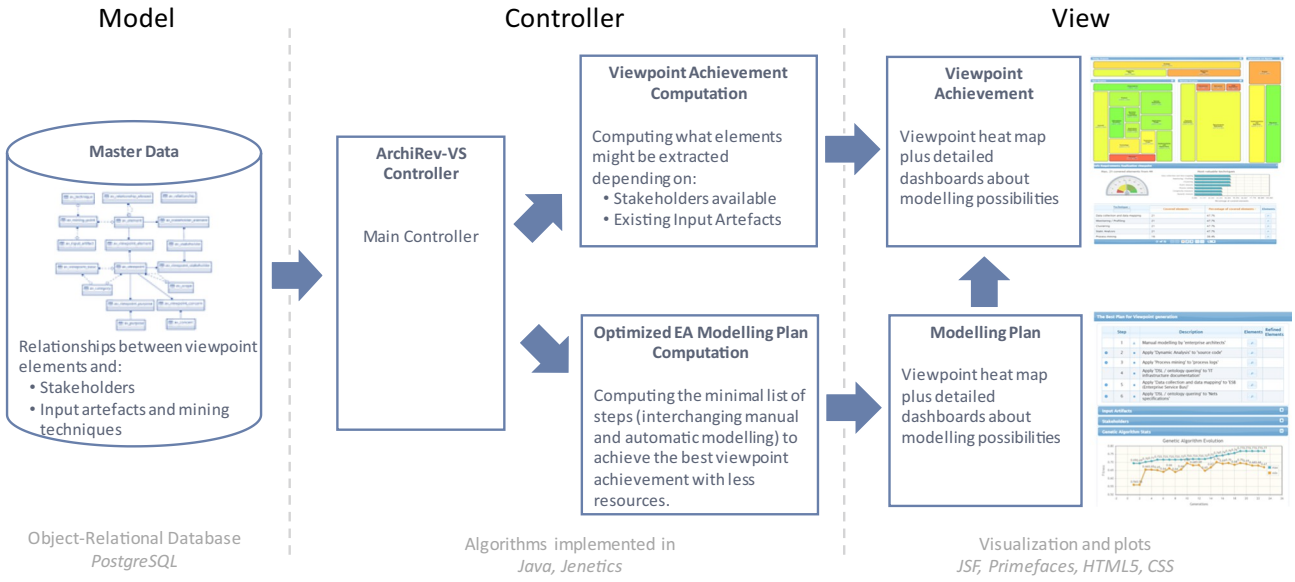


Fig. 4. Software architecture for ArchiRev-VS.

automatically). A more detailed explanation of this is provided in Section 4.3. The controller layer (implemented in *Java*) defines certain algorithms to provide the two mentioned contributions. First, (see top part of Fig. 4) viewpoint coverage computation that estimate the degree to which viewpoints could be modelled with the selected artefacts and experts. These algorithms search for the relationships mentioned previously and provide several types of elements that could be extracted depending on the stakeholders assigned and/or the available input artefacts. This is explained in Section 4.4. Second, (see bottom part of Fig. 4) the computation of the optimized EA modelling plan for some selected viewpoints and certain constraints about available artefacts and experts. This computation is carried out through a genetic algorithm implemented using *Jenetics* [33], a genetic programming library. This is explained in Section 4.5.

Finally, the view layer visualizes (for both functionalities) a heat map with a certain viewpoint layout containing the estimated percentages. For the optimized plan computation, the main result is a list of modelling steps with different artefacts (together with specific mining techniques) and experts to be used to generate EA viewpoints.

4.3. Master data generation

The cornerstone of this research is the data concerning the relationships between input artefacts and EA viewpoint elements (depending on the mining technique that might be employed), and associations with experts and certain elements that they are able to model for every viewpoint. These relationships have been coded and persisted in a relation database in order to serve as the knowledge base of the system. Fig. 5 shows the relational schema designed for this purpose. The core element in this schema is the entity *av_viewpoint*, from which other relationships are defined.

Table 2 provides a summary of the general information for viewpoints. The first step consisted of capturing the information provided by the ArchiMate specification regarding the definition of viewpoints [14]. ArchiMate defines 23 viewpoints organized into four viewpoint categories (entity *av_category*). Each viewpoint defines certain concerns (*av_concern*) and purposes (*av_purpose* plus the multiple relationship table *av_viewpoint_purpose*) that represent a subset of three possible values: designing, deciding and informing.

Each viewpoint defines a list of ArchiMate elements that may appear in the respective view (see the *av_viewpoint_element* and *av_element* path in Fig. 5). ArchiMate also defines a list of stakeholders (see Table 3) for each viewpoint (represented using the *av_viewpoint_stakeholder* and *av_stakeholder* path).

It is important to highlight that this list of stakeholders merely represents the roles that are related to a certain viewpoint without indicating which of them can model each element involved in the viewpoint. We have, therefore, defined the relationships between stakeholders and the elements that they can manually model for each viewpoint. This is represented in the *av_stakeholder_element* entity (see Fig. 5). This information has been defined by the authors involved in this research during various workshops using a multi-round Delphi study as similar works did [34]. These Delphi meetings consisted of the three authors of this work, with one of these playing the role of facilitator. All links between stakeholders and viewpoints were provided by each participant in an anonymous way, then feedback was provided and consensus through discussions was achieved for every viewpoint and set of stakeholders. Eventually, we established a total of 242 relationships.

In a similar way to that which occurs with relationships between stakeholders and elements in viewpoints, we define relationships for input artefacts as persisted through *av_input_artifact* (see possible values in Table 3), reverse engineering or mining techniques (*av_technique*) and elements (*av_element*) that can be discovered. These relationships, therefore, provide links among three different entities by means of the entity called *av_mining_point* (see Fig. 5). These relationships were designed in the aforementioned workshops based on Delphi technique, although they were based to a great extent on the main insights extracted by means of the systematic mapping study conducted previously [7]. This research defines a total of 781 relationships. Please note that both types of relationships could be customized in each company according to each stakeholder's responsibility and knowledge or depending on the specific techniques/tools that allow further or different elements to be discovered for each viewpoint. These relationships are stored in a relational database and there exist many applications to visualize these relationships in some tables and change it easily by any people independently of their skills. A script with which to build the fully functional database is available at [35].

4.4. Viewpoint coverage computation

ArchiRev-VS can compute the theoretical viewpoint coverage concerning different input artefacts and/or stakeholders that have been chosen previously. The viewpoint coverage (C_V) is defined for each viewpoint v according to the formula in Eq. (1), which capture the idea of how much of the total possible population of elements in a viewpoint might be retrieved. C_V is computed by considering the ArchiMate

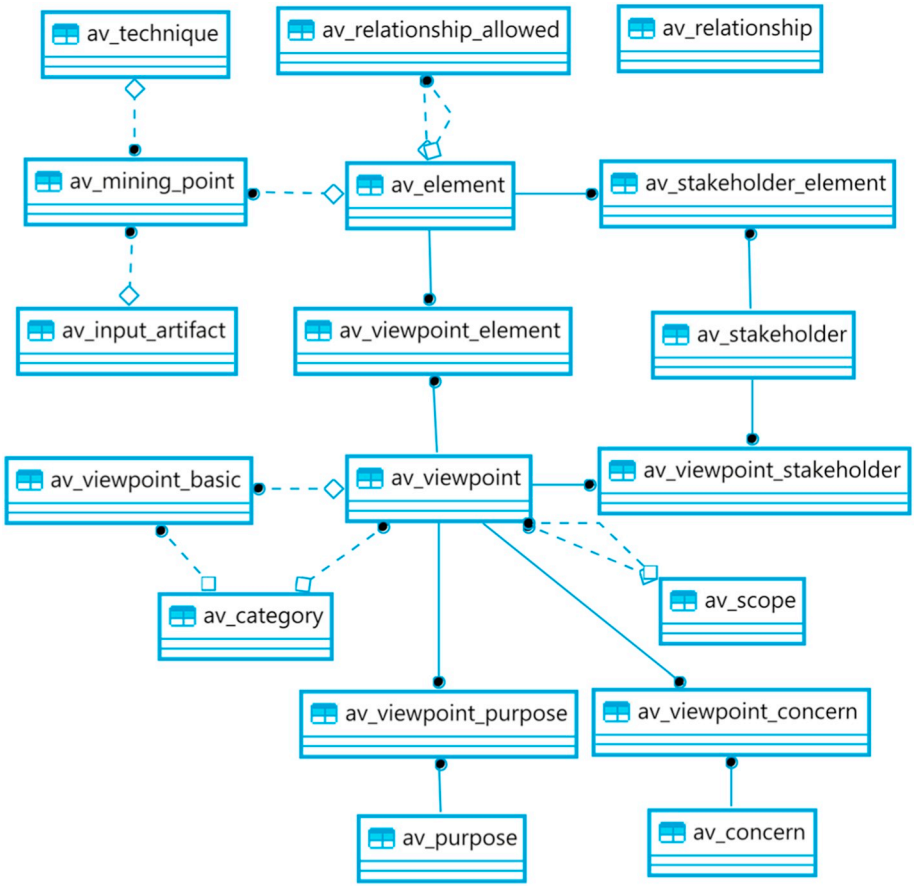


Fig. 5. Relational schema for viewpoints, artefacts, mining techniques, and stakeholders.

Table 2
Overview of ArchiMate viewpoint specifications.

Cat.	Viewpoint	Concerns	Purpose		
			Designing	Deciding	Informing
Basic	Organization	Identification of competencies, authority, responsibilities	•	•	•
	Information structure	Consistency, completeness, structure and dependencies of the data and information used	•		
	Technology	Stability, security, dependencies, costs of the infrastructure	•		
	Physical	Relationships and dependencies of the physical environment and how this relates to IT infrastructure	•		
	Product	Product development, value offered by the products of the enterprise	•	•	
	Application usage	Consistency, completeness, reduction of complexity	•	•	
	Technology usage	Dependencies, performance, scalability	•		
	Business process cooperation	Dependencies between business processes, consistency, completeness, responsibilities	•	•	
	Application cooperation	Consistency, completeness, relationships and dependencies between applications, orchestration/choreography of services, reduction of complexity	•		
	Service realization	Consistency, completeness, added value of business processes, responsibilities	•	•	
	Implementation and deployment	Structure of application platforms and how they relate to supporting technology	•	•	
	Layered	Consistency, reduction of complexity, impact of change, flexibility	•	•	•
Motivation	Requirements realization	Motivation, architecture tactics, architecture strategy	•	•	•
	Stakeholder	Motivation, architecture mission, architecture strategy	•	•	•
	Motivation	Motivation, architecture tactics, architecture strategy	•	•	•
	Goal realization	Motivation, architecture tactics, architecture mission, architecture strategy	•	•	
Strategy	Resource map	Motivation, architecture tactics, architecture strategy	•	•	
	Strategy	Strategy development	•	•	
	Capability map	Motivation, architecture tactics, architecture strategy	•	•	
	Outcome realization	Business-oriented results	•	•	
Implementation and migration	Project	Motivation, architecture vision and policies		•	•
	Implementation and migration	Motivation, architecture vision and policies		•	•
	Migration	History of models	•	•	•

Table 3
Input artefact and stakeholder lists.

Input artefact	Reverse engineering/mining technique	Stakeholder
• IT infrastructure documentation • Sensor/actuator event logs • Management Data Repository (MDR) • Applications networks • Application portfolio • Application architecture • Use cases • Business goals • RDBMS schema • ORM specification • ESB (Enterprise Service Bus) • ESB (Enterprise Service Bus) - BPEL Logs • BPMN models • ArchiMate models • Process logs • Application invocation logs • Operational data (data warehouse) • EA models • Social media info • Server dynamic info (CPU/bandwidth/traffic) • Microservices calls • Services • Business groups • IT operational data • Project management info • LDAP info • Code repository • Version management system • Continuous improvement system • Nets specifications	• Static analysis • Dynamic analysis • Code slicing • MDA transformation • Data collection and data mapping • Process mining • Social network analysis • DSL/ontology querying • Complexity measure • Variability mining • Clustering • Model weaving • Frequent closed sequential pattern • Serious games • Process re-engineering • Business intelligence • Crowd social analysis • Draft elements management • Web data mining • Monitoring/profiling • Design structure matrices • Pattern matching	• Enterprise architects • Process architects • Domain architects • Application architects • Information architects • Infrastructure architects • ICT architects • Employees • Shareholders • Stakeholders • Managers • Operational managers • Business analysts • Business managers • Product managers • Requirements managers • Product developers • Business architects

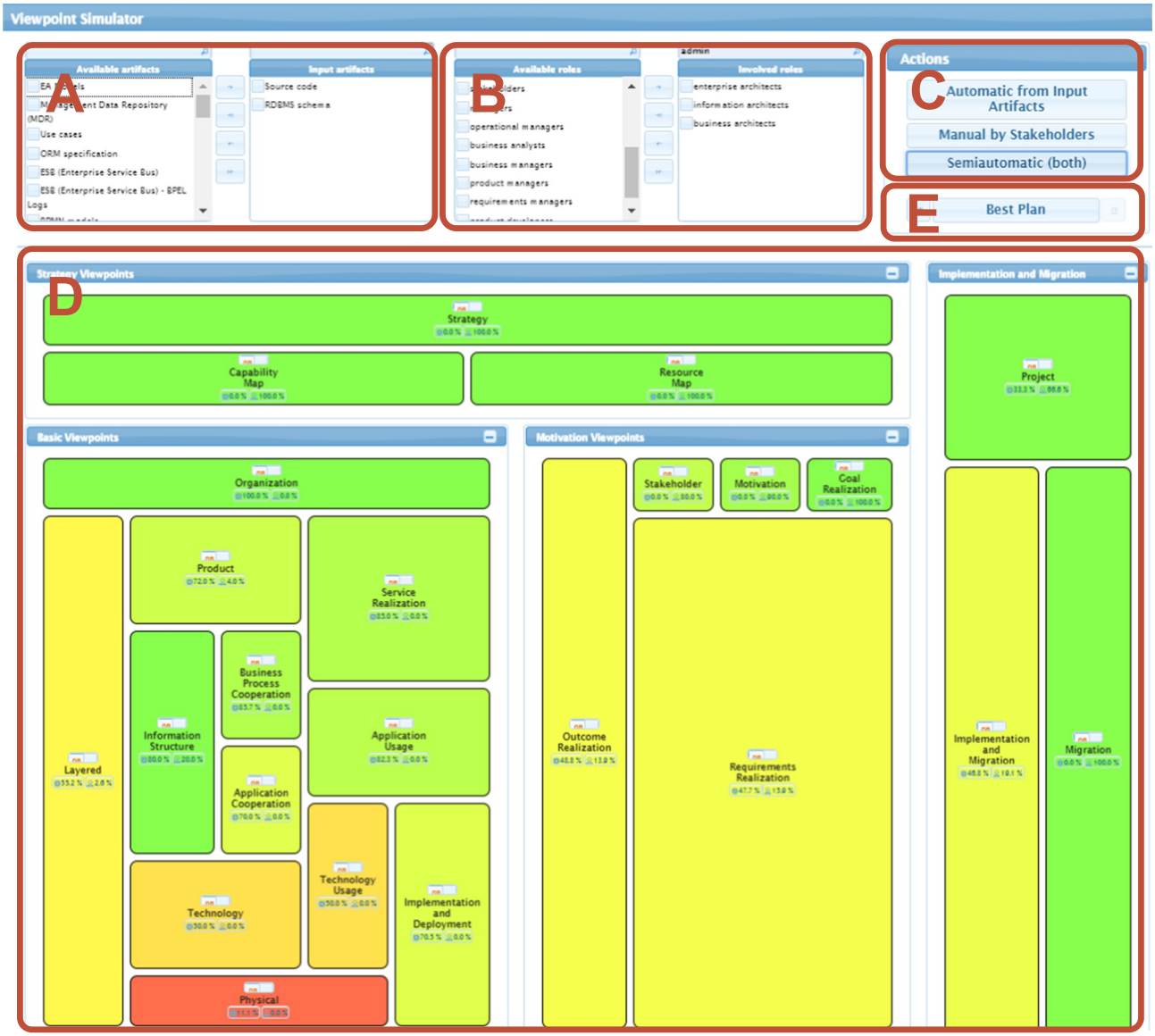


Fig. 6. ArchiRev-VS main user interface with a hybrid (semiautomatic) modelling results.

elements that could be discovered automatically for some of the reverse engineering techniques and their given artefacts, or elements manually modelled by the selected stakeholders. The percentage (see Eq. (1)) is then computed with these elements and divided by the total number of elements (according the ArchiMate standard) that can be used in the viewpoint.

$$C_v = \frac{Elements\ discovered\ automatically\ +\ Elements\ modelled\ manually}{Total\ elements\ for\ v\ (in\ ArchiMate)} \quad (1)$$

The computation of the viewpoint coverage is the key aspect for the algorithms of ArchiRev-VS (see Appendix I). The system accepts as input two different lists with (i) artefacts that might be available in the organization to be used with each mining technique; and (ii) the stakeholders that are active in the organization and that might be assigned to the EA modelling project. These two input lists are the subsets of the elements available for each computation (see Table 3). Having selected this input, there are three possible computations regarding the nature of EA modelling: (i) automatic that considers only input artefacts and mining techniques (see Algorithm 1 in Appendix I), (ii) manual that considers only stakeholders for the computation of the viewpoint coverage (see Algorithm 2 in Appendix I), and finally (iii) hybrid that considers both, i.e., it first prioritizes automatic EA modelling and then computes viewpoint coverage for those elements that it was not possible to discover automatically.

Algorithm 1 calculates the coverage for automatic modelling. For each viewpoint, it computes the elements that can be discovered from the selected artefacts. Together with the viewpoint coverage, Algorithm 1 builds a map that relates input artefacts, techniques and elements. Individual percentages from formula (1) could subsequently be visualized in order to compare how each technique performs for each viewpoint elements. As occurs with Algorithm 1, Algorithm 2 computes both the viewpoint coverage percentage and a map relating stakeholders and elements for each viewpoint. A third algorithm based on the composition of the other two algorithms has been designed for computing viewpoint coverage for hybrid modelling. Overall, this algorithm applies Algorithm 1 and Algorithm 2 in a row with one particularity. Elements that are discovered (during coverage computation for automatic modelling) are not considered during the queries executed in Algorithm 2. Please note that all the algorithms are based on database queries that check relationships that have been established in this research (cf. Section 4.3). Query 1 in Appendix I illustrates one of

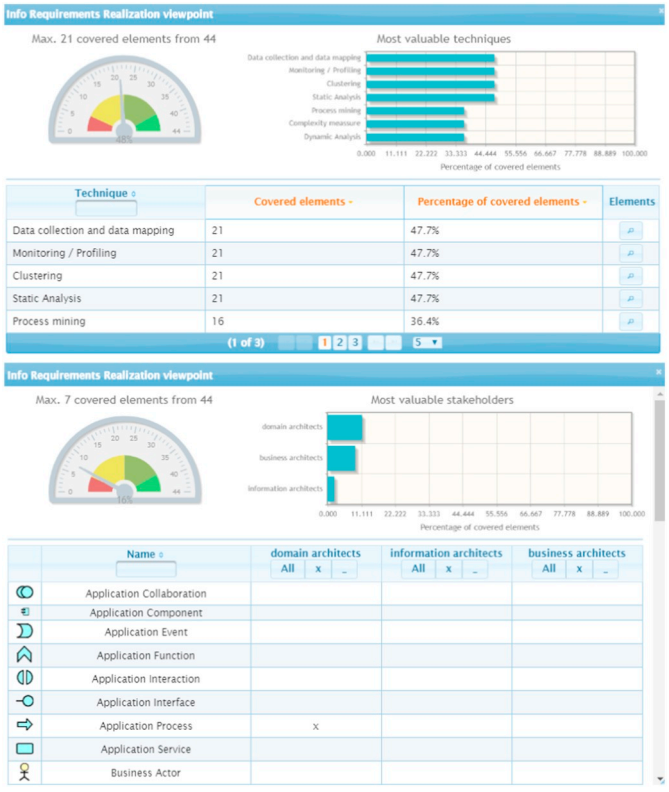


Fig. 7. Viewpoint coverage panels with fine-grained details.

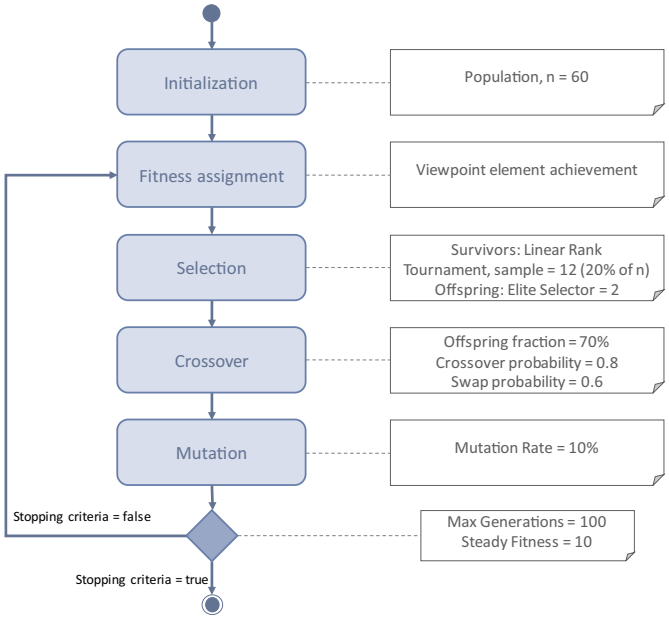


Fig. 8. Genetic algorithm parametrized for computing optimized plans for EA modelling.

these queries when used in Algorithm 1. It is fully operative with the database that can be built using the script available in [35].

Fig. 6 shows the main user interface for the system developed. The snapshot shows the result for a semiautomatic modelling, using source code and RDBMS as input artefacts, and considering domain, information and business architects as stakeholders. The user interface of the tool is divided into 5 main panels (A to E in Fig. 6). The objective of panels A and B is to provide input for coverage computation. These panels consist of two picker lists from which to choose both a subset of artefacts (panel A) and a subset of stakeholders (panel B). It is not necessary to choose elements from both lists. For example, the input artefacts list could be empty when conducting manual modelling computations, for which only a list of selected stakeholders is mandatory. Panel C in Fig. 6 contains the buttons used to perform the three types of computations mentioned previously. Panel E in Fig. 6 performs the computation (through the genetic algorithm) of the best EA modelling plan for the input selected in panels A, B and D.

Panel D in Fig. 6 provides a layout containing the 23 viewpoints considered (see Table 2). This layout arranges viewpoints in four groups according to categories, i.e. strategy (top), basic viewpoints (bottom left), motivation (bottom right), and implementation and migration (right-hand side). It organizes viewpoints according to the category but also as regards the abstraction level and the ArchiMate layer. Thus, for example, physical and technology viewpoints are at the bottom, while the strategy viewpoint is located at the top. After performing one of the three available coverage computations, the viewpoints are coloured as a heat map regarding the viewpoint coverage, C_V (cf. formula (1)). We believe that this visualization (based on a heat map) facilitates the understanding of the situation regarding the whole set of EA viewpoints. Please note that each viewpoint in panel D (see Fig. 6) has two buttons with the C_V percentage computed for the current execution. On clicking these buttons, further panels are shown with further, fine-grained information (see Fig. 7).

On the one hand, the detailed information panel for automatic modelling (top part in Fig. 7) shows the sorted list for all the techniques that are related to the selected input artefacts. Moreover, this panel can visualize the elements that each technique is able to discover. This panel is useful as regards selecting those reverse engineering techniques that are most effective in the current scenario.

On the other hand, the detailed information panel for coverage of manual modelling (bottom part in Fig. 7) shows the stakeholders that are able to model the majority of the elements required in the current viewpoint. This panel also provides a matrix of the stakeholders and elements that could be modelled for everyone.

4.5. Optimized EA modelling plan computation

The second major functionality of ArchiRev-VS [32] is the computation of an optimized plan for modelling EA viewpoints. This computation can be calculated for a subset of target viewpoints (as selected previously). This is also computed for those input artefacts and experts that are available in a certain company. Even if there are plenty of input artefacts and experts, the modelling plan could be computed for cases in which some input artefacts or experts are not considered.

This computation has been designed and implemented as a genetic algorithm to provide the modelling plan. We have selected genetic algorithm since this problem can be seen as feature selection problem (formulated as a combinatorial problem), in which the best set of pairs of input artefacts and mining techniques, plus experts have to be selected. Feature selections problems consists of the identification of the most relevant features for a predictive model. Irrelevant and redundant features are removed since these do not contribute or even decrease the overall performance of the predictive model. In our case, an exhaustive selection of features (input artefacts, mining techniques, and experts). In particular, our problem consists of 833 features ((37 input artefact × 22 mining techniques = 814 pairs) + 19 stakeholders = 833). The number of all the possible combinations can be calculated with the combinatorial formula in Eq. (2), where n is the number of features and p the number of selected features. This number tends to infinity.

$$\sum_{p=1}^{p<n} C_n^p = \sum_{p=1}^{p<n} \left(\frac{n!}{p!(n-p)!} \right) \rightarrow \infty \quad (2)$$

A genetic algorithm is a “stochastic method for function optimization based on the mechanics of natural genetics and biological

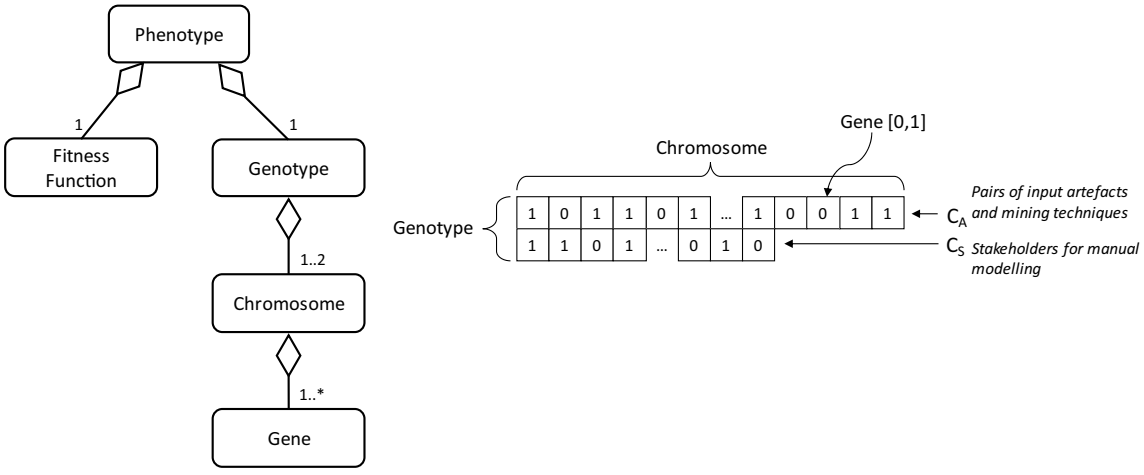


Fig. 9. Data model of genetic algorithms and the specific data structure used.

evolution" [36]. Genetic algorithms follow the rules of natural selection and biological evolution. Genetic algorithms (see Fig. 8) "repeatedly modify a population of individual solutions from an initial population". Then a fitness value is computed for each individual (according to their adaptation to the environment) and some of them are selected to be parents. Those produce offspring where genes are cross over from one individual to another. Some mutation operations are also introduced in this point according to the stochastic nature. Over successive generations, the population evolves toward an optimal solution. The overall structure and steps of a standard genetic algorithm can be seen in left side of Fig. 8. This algorithm is well-known and has been applied to a wide range of domains. Hence, next subsections focus on explaining how this algorithm has been applied and parametrized to compute optimal plans for EA modelling (see right side of Fig. 8). As usual in evolutionary programming, some of these parameters have been optimized by trial and error (cf. Section 5).

4.5.1. Initialization

The population size is the first important parameter for the initialization phase and next generations. Population size refers to the number of individuals (i.e., genotypes) with different genes grouped into chromosomes. Fig. 9 (left side) shows the aggrupation of the data used during the algorithm. At the initialization stage, genes of individuals are randomly generated. In our algorithm, we consider two chromosomes (see right side of Fig. 9): the set of selected pairs of input artefacts and mining technique, and the set of experts. These contains Boolean genes that specify if each input artefact or expert is considered. After initialization, for next generations, population depends on crossover, offspring and mutation phases. Population size influences the overall performance of genetic algorithms. Small populations might not be enough to get the optimal solution (e.g., because of premature convergence in a local optimal). While, larger populations might cause an algorithm to slow down. Extensive research suggests defining the optimal size of population by trial and error [37,38]. We carried out this process and determined an optimal size of 60 (cf. Section 5).

4.5.2. Fitness assignment

After the population have been initialized, the fitness of every individual must be computed. The greater fitness, the greater probability of being selected for recombination. Thus, the core part of the genetic algorithm is its fitness function. In our case we have designed a function that trust on the concept of viewpoint coverage we introduced in previous section (see formula (1)). In general words, what this function does is to compute the number of ArchiMate elements (for each viewpoint) that are able to be modelled using the input artefacts, mining techniques and experts represented in genes of each individual. Nevertheless, this function computes three different concepts (*AM – Average Mining*, *CV – Completed Viewpoints*, and *UE – Unused Elements*) for the three kind of mining (automatic (a), manual (m) and hybrid (h)) which are grouped in the single formula (3).

$$F(G) = W_{a1}\cdot AM_a + W_{m1}\cdot AM_m + W_{h1}\cdot AM_h + W_{a2}\cdot CV_a + W_{m2}\cdot CV_m + W_{h2}\cdot CV_h + W_{a3}\cdot UE_a + W_{m3}\cdot UE_m + W_{h3}\cdot UE_h \quad (3)$$

Table 4
Possible configurations for computing optimized EA modelling plans.

Parameters												
	Modelling	Prioritize automatic	•		•							
		Prioritize manual		•		•						
		Best					•		•		•	
	Performance	Maximize overall performance	•			•			•			
		Maximize completed viewpoints		•			•					•

Component	Weight	Values					
AM_a	W_{a1}	0.35					
AM_m	W_{m1}			0.35			
AM_h	W_{h1}	0.15		0.15		0.50	
CV_a	W_{a2}		0.35				
CV_m	W_{m2}				0.35		
CV_h	W_{h2}		0.15		0.15		0.50
UE_a	W_{a3}	0.15	0.15	0.15	0.35		
UE_m	W_{m3}	0.35	0.35	0.35	0.15		
UE_a	W_{h3}					0.50	0.50

Combinations of these expressions lead to fitness function with 9 values with their respective weights. In following paragraphs these three expressions are explained and illustrated through the formula for automatic modelling. Similar expressions are used for manual and hybrid modelling.

AM (Average Mining) represents the percentage of elements extracted for each viewpoint on average. Formula (4) illustrates AM_a as defined for automatic extraction based on pairs $\{i,t\}$ of the Cartesian product of input artefacts, *I*, and mining techniques, *T*. This is computed regarding each viewpoint *v* in *V*, the set with the selected EA viewpoints. Technically, the number of ArchiMate elements covered in each case are computed with a database query similar to Query 1. Possible decimal values of formula (4) ranges from 0 to 1 and represent the percentage in which the selected viewpoints can be modelled. In a similar manner to *AM*, *CV* (Completed Viewpoints) computes the percentage of viewpoints fully completed, i.e., for which *AM* is 1. Formula (5) computes this concept for automatic modelling. The third component, *UE* (Unused Elements) represents the number of unused pairs of input artefacts and mining techniques as well as unused stakeholders in manual modelling. Roughly, *UE* is the proportion of genes that are zero within the total number of genes in the chromosome (see formula (6)).

$$AM_a = \frac{\sum_v^V \left| \bigcup_{\{i,t\}}^{I \times T} (Elements\ covered\ per\ \{i,t\}\ in\ v) \right|}{|V|} \tag{4}$$

$$CV_a = \frac{\sum_v^V \left(1 - \min \left\{ 1, |V| - \left| \bigcup_{\{i,t\}}^{I \times T} (Elements\ covered\ per\ \{i,t\}\ in\ v) \right| \right\} \right)}{|V|} \tag{5}$$

$$UE_a = \frac{\sum_g^{C_A} C_A(g)}{|C_A|},\ C_A(g) \in \{0,1\} \tag{6}$$

The rationale of this fitness function is that the solution proposed follows a minimax approach, i.e., this function tries to maximize the viewpoint coverage while resources are minimal. Thus, it allows modeling as more viewpoints as possible, while this is achieved with minimal, non-overlapped resources, i.e., reducing cost for companies. Additionally, together with the algorithm, the fitness function is parametrizable by considering 9 different weights for the previous expressions. Such weights changes in every configuration according to the values presented in Table 4. Parametrization consists of two parameters. First, the prioritization for one of the three types of modelling: automatic, manual or hybrid. Second, the maximization strategy. This strategy might be focus on the overall performance (i.e., total number of covered elements in accordance to *AM*), or it may be focus on maximizing fully completed viewpoints (in accordance to *CV*). These two parameters, with three and two values respectively, lead to 6 possible combinations. These combinations can be selected by the user by clicking in the configuration button to the left of the button labelled as 'Best Plan' in Panel E (see Fig. 6).

4.5.3. Selection

In this phase, those genotypes more fitted to the environment are recombined for the next generation. We distinguish two different selections: survivor and offspring selection. First, survivor selection indicates what individuals survive to the next generation. In this case, we chose a linear rank selector. This is similar to a roulette-wheel selector, in which the selection probability is proportional to the fitness of individuals. However, linear rank selector fix problems when the fitness values differ very much. If the best genotype fitness is too high, e.g. 90%, then other genotypes have a neglectable probability to be selected [39]. Thus, linear-ranking selection sorts individuals according to their fitness values and the selection probability for each individual is linearly assigned.

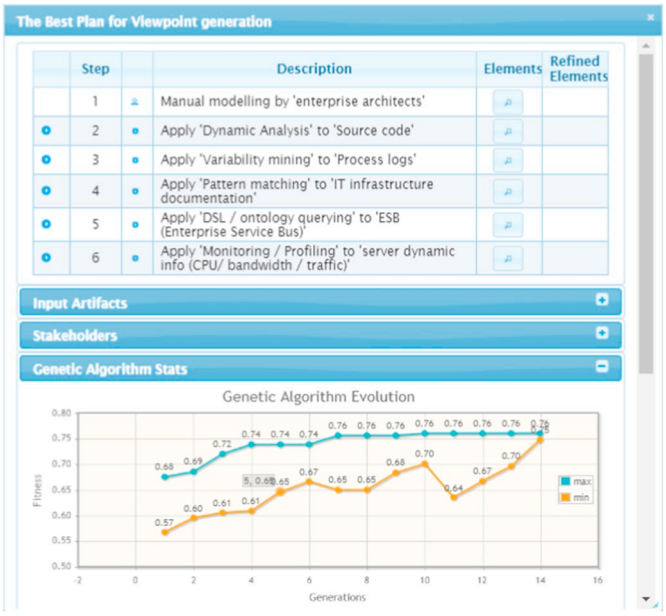


Fig. 10. Optimized EA modelling plan provided by ArchiRev-VS.

After this selector, offspring selector is then used for selecting the offspring population. In this case we used a tournament selector. It takes samples of two or more individuals and chooses the individual with greater fitness value for each sample. The worst genotype will never survive while the best one will win all tournaments in which is involved. The size s of samples must be determined. For larger samples, weak genotypes have a neglectable chance of being selected. On the other hand, small samples might lead to local optimal results. In our algorithm we establish 20% of the population size. After algorithm optimization (cf. Section 5) the population size was established in 60, as a result, the size of tournament selector samples is defined as 12.

Regarding linear rank, that is a fitness proportional selectors, "the tournament selector is often used in practice because of its lack of stochastic noise. Tournament selectors are also independent to the scaling of the genetic algorithm fitness function" [33]. In addition to the tournament selection, we consider elitism selection that makes the fittest individuals to survive directly for the next generation, i.e., with no recombination. This is typically used in genetic algorithm to improve performance, since it does not spent time re-discovering better solutions found so far. The problem is that the algorithm might converge to a local optimum. As a result, the number of genotypes selected by elitism must be low. We select the 2 best individuals in each generation.

4.5.4. Crossover

The crossover phase recombines individuals previously selected to generate a new population. This phase takes two individuals at random and combines their genes into new individuals, until the new population as the target size. The algorithm we have designed uses a uniform crossover, which swaps genes between parent chromosomes with a certain swap probability. Uniform crossover is a more exploitative approach, i.e., it provides a better search of the design space, than other approaches that maintains whole fragments of chromosomes [40]. This phase is parametrized by three different factors. Section 5 shows how these values were optimized by trial and error.

- Offspring fraction defines the percentage of offspring to be evaluated in the next generation. This is established at 0.7.
- Crossover probability determines if a crossover will happen, otherwise, the offspring are identical to the parents. This is established to 0.8.

Table 5
Overview of different configurations for the optimization of the genetic algorithm.

		Optimized parameters		
		RQ 2.1. Population size	RQ 2.2. Offspring fraction	RQ 2.3. Crossover/swap probability and mutation rate
Configuration	Priority	Automatic, manual, hybrid	Automatic, manual, hybrid	Only hybrid
	Population size	?	60	60
	Elitism	2	12 (20%)	12 (20%)
	Tournament sample	2	2	2
	Offspring fraction	0,6	?	0,7
	Crossover probability	0,2	0,2	?
	Swap probability	0,2	0,01	?
	Mutation rate	0,01	0,2	?
	Maximum generations	∞	100	100
	Steady fitness limit	10	10	10

- Swap probability determines the probability with which genes are swapped during crossover. This is established to 0.6.

4.5.5. Mutation

Despite the crossover phase, the new population could present low diversity regarding their ancestors. As a result, the mutation phase is aimed at improving diversity and, consequently, extending the search space and avoiding the mentioned local optimums. Mutation changes the value of individual genes with a certain probability. The mutation probability is usually low, but this value depends on how crossover phase was designed. If mutation is the only source of exploration, mutation probability should be higher. The mutation probability was established to 0.1 by trial and error (cf. Section 5).

4.5.6. Generations and solution presentation

The previous phases are iteratively executed (see Fig. 8) until a stopping criterion is met. Our criteria consist of two conditions, a maximum number of generations and the steady fitness strategy, which finishes the evolution if the best fitness hasn't changed after a given number of generations. The maximum number of generations is defined as 100. During the optimization of the population size, this value was set up to the maximum number, i.e., only the steady fitness limit was considered in practice. In such scenario, the mean of generations was between 33 and 37 while the maximum generations for one of the cases were 69. As a result, we believe 100 generation at maximum is reasonable. Regarding the steady fitness limit, it was established at 10. It is reasonable to think that crossover nor mutation can generate a better population after 10 generations obtaining the same fitness.

The output of the genetic algorithm is a genotype with two chromosomes with two sets of genes (see Fig. 9), representing two sets of (i) pairs of input artefacts and mining techniques for automatic modelling, and (ii) stakeholders for manual modelling. The information of the best genotype is decoded and presented in ArchiRev-VS once the execution of the genetic algorithm is completed. The top panel in Fig. 10 provides the two mentioned sets together with the modelling order according to its contribution to the overall viewpoint coverage. For every step, the system provides the set of elements that should be modelled according to the knowledge of the expert or regarding the capabilities of the mining technique. Also, the 'Refined elements' column shows the elements a stakeholder (who did a manual modelling of specific viewpoints in a previous step) is able to refine in a following step regarding an automatic modelling. In this way, a stakeholder can be alerted to review the work done automatically for some viewpoints. The bottom panel in Fig. 10 provides the evolution of the genetic algorithm throughout the different generations. In this plot, max and min represent the maximum and minimum fitness values achieved for individuals of the population in each generation.

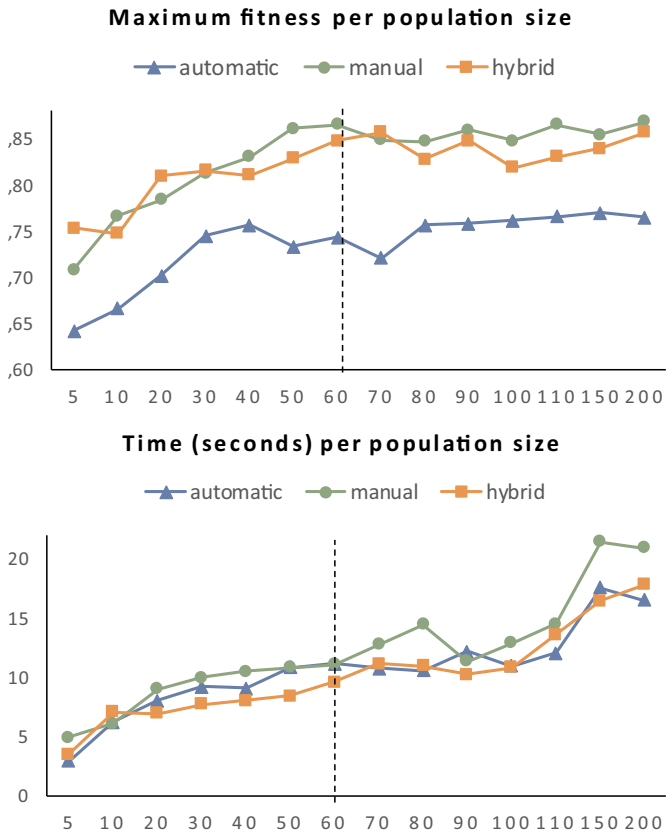


Fig. 11. Optimization results for population size.

5. Validation and genetic algorithm optimization

This section presents the validation and optimization of the genetic algorithm. This validation has been planned, designed and conducted based on the Goal/Question/Metric (GQM) approach.

5.1. Goals, research questions and metrics

The *object of study* is ArchiRev-VS, the tool designed and developed and, in particular, the genetic algorithm. While, the *purpose of the study* is twofold. First, Goal 1 tries to demonstrate the feasibility of the proposed system to figure out optimal EA modelling plans. According to DSRM, this kind of validation focuses on assessing if the artefact fulfils the goals for which it was designed and build. In our case, it tries to demonstrate that ArchiRev-VS can provide EA modelling plans in a reasonable time. We are conscious that further, stronger empirical validation is necessary to determine if the provided EA modelling plans

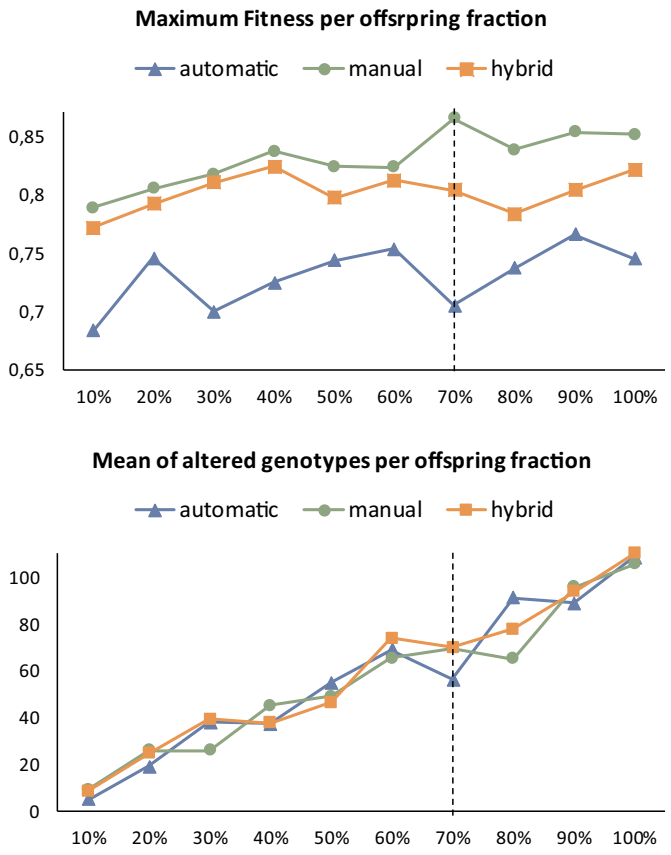


Fig. 12. Optimization results for offspring fraction.

are useful for real companies and its execution lead to success in terms of quality of EA models and modelling costs. This kind of validation must be done in a long-term period and it is key to conduct first the kind of validation proposed in this paper in order to appeal companies to use ArchiRev-VS. Then, Goal 2 is aimed to present how the genetic algorithm has been optimized.

Goal 1. Demonstrate feasibility of ArchiRev-VS

RQ 1.1. Does ArchiRev-VS produce EA modelling plans?
RQ 1.2. Do the EA modelling plans provide optimum sets of input artefacts and experts that are combined for automatic and manual modelling?

Goal 2. Optimize the performance of the genetic algorithm in ArchiRev-VS

RQ 2.1. How does population affect the fitness value?
RQ 2.2. How does offspring fraction affect the fitness value?
RQ 2.3. How do crossover & mutation rate affect the fitness value?
RQ 2.2. What is the time spent by genetic algorithm to compute EA modelling plans?

The main output metrics are fitness value (see formula (3) in Section 4.5.2), which is produced by the genetic algorithm as well as the execution time. On the other hand, the input metrics are those used to parametrize the genetic algorithm. Different configurations are presented in Table 5.

5.2. Population effect analysis

The population size is one of the most critical success factors in evolutionary algorithms. This optimization was made, as other studies [37,38], by trial and error. It consists of testing several population sizes with a fixed configuration and then evaluates various characteristics. In case of population size, the analysed characteristics are maximum fitness value per generation as well as the execution time. It is well known that greater populations might lead to low performance algorithms without better results.

The remaining parameters were fixed as depicted in Table 5. Columns represent target parameters to be optimized, while rows indicate how other parameters where fixed during optimization. In this case, these values are taken from the same range than other genetic algorithms defined.

During all optimizations, three different executions are considered according to the priority parameter for the fitness function, i.e., automatic, manual and hybrid, since this factor might influence the maximum fitness value per generation as well as other measured characteristics. Fig. 11 shows the results for the population optimization after testing for different sizes. The top part in Fig. 11 shows the maximum fitness value achieved for each execution, which present an upward trend up to size of 60 individuals. After that, the three cases present an asymptotic behaviour. The bottom part in Fig. 11 shows the average of time spent per generation. Time values are normalized regarding the number of generations to be comparable for all the different population sizes. Time present a slightly upward trend, although it dramatically grows for sizes higher to 100 individuals. Since no better results are achieved with sizes higher to 60, there is no reason for limiting the overall performance. As a result, the population size is established in 60 individuals.

5.3. Offspring fraction effect analysis

Another parameter to be optimized is the offspring fraction (O_f), that is used during crossover phase. This is defined as the percentage of offspring to be evaluated in the next generation. While, $1\text{-}O_f$ is the proportion of the selected survivors for the next generation. Different values (0.1, 0.2, …, 1) were tested with the configuration shown in Table 5. We used the 'default' setup and a population size of 60 as optimized previously. Fig. 12 presents results for all executions performed.

The analysis of the best fitness achieved for each execution shows that there are not significant differences. Meanwhile, genotypes altered on average grow significantly (see bottom plot in Fig. 12), which is also associated with a greater computational time. Despite there are no huge differences in fitness values, altering ratio must be high to explore the search space and avoid local optimums. Hence, we chose 70% as the optimum offspring fraction.

5.4. Crossover & mutation effect analysis

Crossover and swap probabilities are also optimized. In this case, these two values are analysed in combination with the mutation rate since these three values are all together an impact in the overall diversity of genotypes and, thus, its influence on fitness should not be assessed in isolation. The optimization was conducted with several executions with some fixed parameters according to Table 5, as well as through the execution with different values for the three parameters under evaluation. In this case, hybrid modelling was the only prioritization values selected, since it was observed in previous optimization tasks that the fitness values kept the same trend for the three different prioritization values.

Fig. 13 shows the results for this optimization. The three box plots to the left show the distribution of maximum fitness achieved for each execution, while plot to the right show the respective evolution time on average (i.e., normalized for the number of generations). Crossover probability shows a slightly gain when it is higher, although its time grows dramatically. The fitness for swap probability follows a curve with a maximum in 0.6. In this case, although the evolution time follows an upward trend, the grow is moderated with regard to the crossover probability. Finally, mutation rate provides better results up to 0.1, and then the fitness is worst even when the time increases

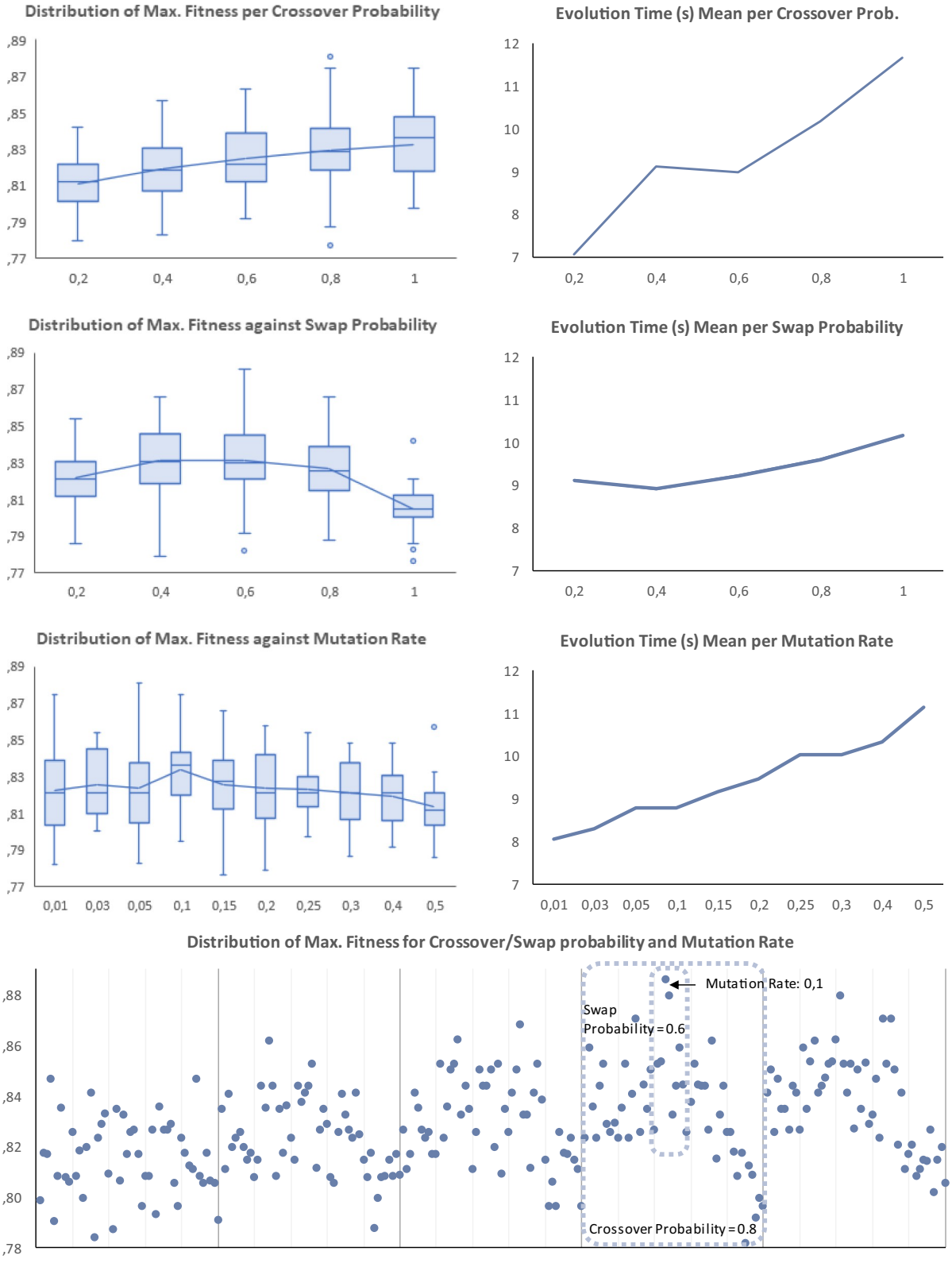


Fig. 13. Optimization results for crossover and swap probabilities and mutation rate.

dramatically. This phenomenon happens because higher mutation rates can lead to mutate those genes that contribute in a greater extent with the optimal solution and move to local optimums instead. Bottom part of Fig. 13 also provides a scatter plot for the combination of the three parameters under study. It can be noticed that the best combination is achieved for crossover probability of 0.8, swap probability of 0.6, and mutation rate of 0,1.

5.5. *Execution time analysis*

Finally, we analysed the computation time of the genetic algorithm. Instead of optimizing the genetic algorithm, the goal of this analysis is to assess if the time for computing optimized EA modelling plans is reasonable and also the proposal is scalable for context with more inputs.

The analysis consisted of recurrent executions for three different

Table 6
Average of time (s) for the three configurations and its relative weight in the total time.

	Selection	% Selection	Altering	% Altering	Evaluation	% Evaluation	Total evolve time
Small	0.002	0.011%	0.005	0.036%	14.251	99.952%	14.258
Medium	0.005	0.012%	0.003	0.007%	43.145	99.980%	43.154
Large	0.004	0.004%	0.007	0.007%	102.750	99.988%	102.762

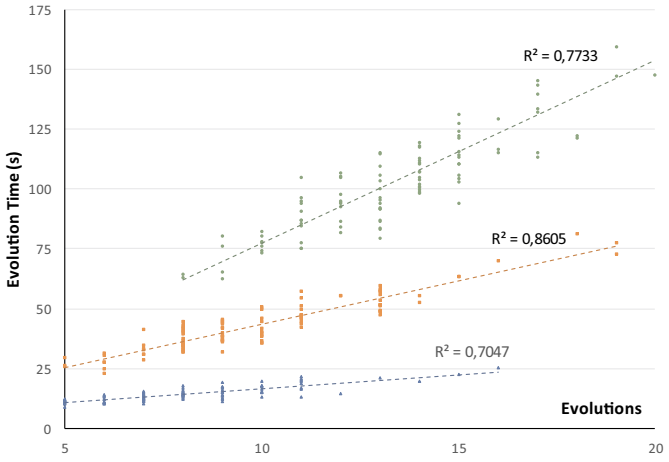


Fig. 14. Time analysis for the three different configurations of input.

inputs (100 executions for each). The goal is to analysis the behaviour of the genetic algorithm with input of different size, i.e., different selected input artefacts and stakeholders that lead to three different lengths of chromosomes. Thus, we tested with large, medium and small input configurations, with respectively 37-19, 18-9, and 9-5 input artefacts-stakeholders. These configurations approximately correspond with the 100%, 50% and 25% of the total available inputs.

The genetic algorithm was parametrized as it was previously optimized, except for the steady fitness limit that was reduced from 10 to 5 for limiting unnecessary computations.

Table 6 shows the computation time on average (i.e., mean values for the 100 executions) for the main phases of the genetic algorithm (selection, altering and fitness evaluation). About the total time, small input can be computed in 14 s on average, while larger input needs around a minute and a half, since the search space is higher for larger chromosomes. Regarding relative weights of every phase, the most time-consuming is the fitness evaluation phase with 99.9%. Actually, selection and altering time are neglectable in comparison. This is explained due to the fitness value is based on the viewpoint coverage computation (cf. Section 4.5.2) that is, in turn, based on database queries. Although database queries seem the bottle neck of the system, it might be optimized for future releases since this issue is more associated with the usage of technology rather than a structural or bad design of the genetic algorithm. Even so, we believe total time is reasonable for a decision-making support system that explores thousands of combinations.

Fig. 14 shows the scatter plot for the total evolution time regarding the number of evolutions. This plot presents results for the three configurations of input (small, medium and large) and their 100 executions. Additionally, the trend lines are shown together with the correlation values (R^2) for hypothesised statistical linear relationships. The three correlation values are close to 1, hence we can state the linear correlation exist between the number of evolutions and time. Additionally, the number of evolutions is directly related to the amount of selected inputs (the number of genes). As a result, we can expect the total time does not grow in an exponential manner for bigger search spaces, and the scalability of the system is ensured.

6. Conclusions

The main contribution of this research is the design and implementation of ArchiRev-VS, A decision-making support system for EA Modelling. This system allows computing how different input artefacts and assigned experts can help enterprise architects to understand how each artefact or expert contributes to achieving each EA viewpoint. The novelty of this work is that this system computes optimum EA modelling plans through a genetic algorithm. This computation can be parametrized. Enterprise Architects could define some prioritizations, e.g., reduce EA modelling costs, maximize EA model precision, automate as many viewpoints as possible, among other goals.

The relevance of this work is that the proposed system can help enterprise architects with a difficult decision: which input artefacts (plus the respective reverse engineering technique) they should employ in automatic EA modelling, and which experts could be assigned to manual modelling. The proposal copes with this by providing enterprise architects with some knowledge about the capability of those artefacts and experts as regards modelling different EA viewpoints. The main implications of this research are the following:

- It provides a catalogue (in a relational database) of mining techniques plus input artefacts that can be used, together with ArchiMate elements that can be discovered and modelled from them. The main implication is that it is the first attempt to integrate all mining efforts made in the EA modelling field to date, signifying that this catalogue could be used for further research in this field.
- It provides a similar (what/who) catalogue regarding the ArchiMate elements that specific experts can model. Although this catalogue could be viewed as a collection of guidelines, it could also be customized to each company as regards available roles. The catalogue could, on the contrary, additionally be used by organizations in order to discover what the missing roles that they should establish are in order to ensure an effective EA capability.
- The proposed decision-support system aids EA architects for making more informed decisions. On the one hand, the extra costs owing to over assignments in manual modelling could be saved by reducing the number of experts. On the other hand, most optimal reverse engineering and input artefacts could be chosen to achieve accurate and complete EA models to avoid unnecessary manual refinements by experts afterwards. Enterprise architects can be trained with the proposed system and make better decisions to improve the (semi) automatic EA modelling and the continuous adaptations required.

Despite its benefits, this proposal has some limitations. The main limitation is the way in which the viewpoint coverage is computed (see formula (1)). It trusts on types of ArchiMate elements, which might be completed in the future with knowledge concerning ArchiMate relationships among those elements in order to mitigate this threat. Moreover, although we have defined a connection between input artefacts and ArchiMate elements, the actual accomplishment of each reverse engineering/mining technique depends on many other factors, e.g., how the technique is implemented, configured, etc. This information should be investigated in the future. The same rationale is applicable to manual modelling by experts. However, please bear in mind that as a decision-support system, ArchiRev-VS provides a certain level of confidence since it provides a good approximation with which

to make informed decisions.

As mentioned, this research is part of a long-term investigation using DSRM and certain research tasks are planned or under development. We have planned to conduct some case studies in real companies in order to model EA viewpoints regarding some EA modelling plans carried out through ArchiRev-VS. This kind of validation is aimed at assessing whether or not computed and actual values are correlated after modelling, as well as to measure satisfaction of companies and the perceived value of the system by companies.

Author contributions

- **Ricardo Pérez-Castillo**: Conceptualization, Methodology, Software, Validation, Investigation, Writing - Original Draft
- **Francisco Ruiz**: Writing – Investigation, Review & Editing, Supervision, Funding acquisition
- **Mario Piattini**: Writing – Investigation, Review & Editing, Project administration, Funding acquisition

Acknowledgments

This study has been partially funded by the G3SOFT (SBPLY/17/180501/000150), GEMA (SBPLY/17/180501/000293) and SOS (SBPLY/17/180501/000364) projects funded by the 'Dirección General de Universidades, Investigación e Innovación – Consejería de Educación, Cultura y Deportes; Gobierno de Castilla-La Mancha'. This work is also part of the BIZDEVOPS-Global (RTI2018-098309-B-C31) project, Ministerio de Economía, Industria y Competitividad (MINECO) & Fondo Europeo de Desarrollo Regional (FEDER).

Appendix I. Algorithms and queries

This appendix shows algorithms and queries used to compute EA viewpoint coverage.

Algorithm 1. Viewpoint coverage for automatic modelling

```
    input: selectedArtefacts
    output: Cv, viewpointMap
 1  viewpointMap ← { Ø , Ø , Ø }
 2  for each viewpoint in all ArchiMate Viewpoints
 3    extractedElements ← Ø
 4    for each artefact in selectedArtefacts
 5      elements ← query: elements of the viewpoint that
            can be extracted from the given artefact
 6      extractedElements ← elements - extractedElements
 7      techniqueMap ← { Ø , Ø }
 8      for each element inelements
 9        techniques ← query: techniques with which the element
              can be extracted from the given artefact
10        techniqueMap ← {element, technique}
11      viewpointMap ← {viewpoint, artefact, techniqueMap}
12    Cv ← extractedElements.size / viewpoint.elements.size
```

Algorithm 2. Viewpoint coverage for manual modelling

```
    input: selectedStakeholders
    output: Cv, viewpointMap
 1  viewpointMap ← { Ø , Ø }
 2  for each viewpoint in all ArchiMate Viewpoints
 3    extractedElements ← Ø
 4    for each stakeholder in selectedStakeholders
 5    elements ← query: elements of the viewpoint that
            can be modelled by the given stakeholder
 6    extractedElements ← elements - extractedElements
 7    stakeholderMap ← { stakeholder , {elements} }
 8    viewpointMap ← {viewpoint, stakeholderMap }
 9    Cv ← extractedElements.size / viewpoint.elements.size
```

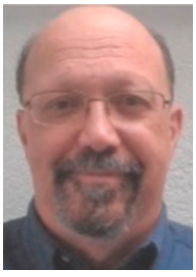
Query 1. Master data query employed in coverage computation based on artefacts

```
SELECT
    v.id, v.name as viewpoint, t.name as technique,
    count(distinct e.name) as num_element,
    (select count(e2.name) as num_elements
      from av_viewpoint as v2, av_viewpoint_element as ve2,
      av_element as e2 where v2.id = ve2.viewpoint_id and
      ve2.element_id = e2.id and v2.id = v.id ) as total_element,
    (cast(count(distinct e.name) as real)*100) / (
      select count(e2.name) as num_elements from av_viewpoint as v2,
      av_viewpoint_element as ve2, av_element as e2 where
      v2.id = e2.viewpoint_id andve2.element_id = e2.id and
      v2.id = v.id ) as percentage
FROM
  av_viewpoint as v, av_viewpoint_element as ve, av_element as e,
  av_mining_point as m, av_input_artifact as a, av_technique as t
WHERE v.id = ve.viewpoint_id and ve.element_id = e.id and
  m.element_id = e.id and m.input_id = a.id and a.id in (:artefacts)
  and m.technique_id=t.id
GROUPBY v.id, viewpoint, technique
ORDERBY viewpoint, percentage desc;
```

References

[1] Gartner, Enterprise Architecture (EA), https://www.gartner.com/it-glossary/enterprise-architecture-ea/, (2018).
[2] S. Kotusev, M. Singh, I. Storey, Consolidating enterprise architecture management research, 2015 48th Hawaii International Conference on System Sciences, 2015, pp. 4069–4078.
[3] The Open Group, TOGAF® Worldwide, [cited 2018; Available from], 2016. http://www.opengroup.org/subjectareas/enterprise/togaf/worldwide.
[4] A. Zimmermann, D. Jugel, R. Schmidt, C. Schweda, M. Möhring, Collaborative decision support for adaptive digital enterprise architecture, BIR Workshops, 2015.
[5] P. Drews, I. Schirmer, B. Horlach, C. Tekaat, Bimodal enterprise architecture management: the emergence of a new EAM function for a BizDevOps-based fast IT, 2017 IEEE 21st International Enterprise Distributed Object Computing Workshop (EDOCW), 2017.
[6] K. Chasioti, BizDevOps: a process model for the alignment of DevOps with business goals, Master Thesis, Business Informatics Department of Information and Computing Science, Utrecht University, 2019, p. 104.
[7] R. Perez-Castillo, F. Ruiz-Gonzalez, M. Genero, M. Piattini, A systematic mapping study on enterprise architecture mining, Enterprise Information Systems 13 (5) (2019) 675–718.
[8] M. Farwick, Towards automation of enterprise architecture model maintenance, in: I. Mirbel, B. Pernici (Eds.), 24th International Conference on Advanced Information Systems Engineering (CAiSE'12) - Doctoral Consortium, 2012, pp. 1–11 Gdansk, Poland.
[9] R. Perez-Castillo, F. Ruiz, M. Piattini, C. Ebert, Enterprise architecture, IEEE Software 36 (4) (2019) 12–19.
[10] M.v.d. Berg, H.v. Vliet, Enterprise architects should follow the money, 2014 IEEE 16th Conference on Business Informatics, 2014, pp. 135–142 Geneva, Switzerland.
[11] P. Johannesson, E. Perjons, An Introduction to Design Science, Springer International Publishing, Switzerland, 2014, p. 197.
[12] K. Peffers, T. Tuunanen, C.E. Gengler, M. Rossi, W. Hui, V. Virtanen, J. Bragge, The design science research process: a model for producing and presenting information systems research, Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), 2006 (sn).
[13] R.J. Wieringa, What is design science? Design Science Methodology for Information Systems and Software Engineering, Springer Berlin Heidelberg, 2014, pp. 3–11.
[14] The Open Group, The ArchiMate 3.0 Enterprise Architecture Modeling Language, http://www.opengroup.org/subjectareas/enterprise/archimate-overview, (2016).
[15] A. Barbosa, A. Santana, S. Hacks, N. von Stein, A taxonomy for enterprise architecture analysis research, Proceedings of the 21st International Conference on Enterprise Information Systems, vol. 2, ICEIS, 2019, pp. 493–504.
[16] M.W.A. Steen, D.H. Akehurst, H.W.L.t. Doest, M.M. Lankhorst, Supporting viewpoint-oriented enterprise architecture, Proceedings. Eighth IEEE International Enterprise Distributed Object Computing Conference, 2004, EDOC 2004, 2004, pp. 201–211.
[17] C. Atkinson, C. Tunjic, Towards orthographic viewpoints for enterprise architecture modeling, 2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, 2014, pp. 347–355.
[18] S. Hacks, M. Brosius, S. Aier, A case study of stakeholder concerns on EAM, 2017 IEEE 21st International Enterprise Distributed Object Computing Workshop (EDOCW), 2017, pp. 50–56.
[19] I. Puspitasari, Stakeholder's expected value of enterprise architecture: an enterprise architecture solution based on stakeholder perspective, 2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA), 2016, pp. 243–248.
[20] D.M. Mezzanotte, J. Dehlinger, Developing and building a quality management system based on stakeholder behavior for enterprise architecture, 15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2014, pp. 1–6.
[21] M. Brosius, S. Aier, M.K. Haki, Introducing a coordination perspective to enterprise architecture management research, 2017 IEEE 21st International Enterprise Distributed Object Computing Workshop (EDOCW), 2017, pp. 71–78.
[22] R.F. Monger, Managerial Decision Making with Technology: Highlights of the Literature, vol. 45, Elsevier, 2014.
[23] J.R. Romero, A. Vallecillo, Well-formed rules for viewpoint correspondences specification, 2008 12th Enterprise Distributed Object Computing Conference Workshops, 2008, pp. 441–443.
[24] M. Ruiz, J. Moreno, B. Dorronsoro, D. Rodriguez, Using simulation-based optimization in the context of IT service management change process, Decision Support Systems 112 (2018) 35–47.
[25] G. Zapata, J. Murga, C. Raymundo, F. Dominguez, J.M. Moguerza, J.M. Alvarez, Business information architecture for successful project implementation based on sentiment analysis in the tourist sector, Journal of Intelligent Information System 53 (3) (2019) 563–585.
[26] F. Kitsios, M. Kamariotou, Business strategy modelling based on enterprise architecture: a state of the art review, Business Process Management Journal 25 (4) (2019) 606–624.
[27] D. Alfonso-Robaina, J.C. Díaz-Moreno, A. Malleuve-Martınez, J. Medina-Moreno, C. Rubio-Manzano, Modeling enterprise architecture and strategic management from fuzzy decision rules, Studies in Computational Intelligence, 2020, pp. 139–147.
[28] O. Sohaib, M. Naderpour, W. Hussain, L. Martinez, Cloud computing model selection for e-commerce enterprises using a new 2-tuple fuzzy linguistic decision-making method, Computers and Industrial Engineering 132 (2019) 47–58.
[29] J.E. Van Aken, Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules, Journal of Management Studies 41 (2) (2004) 219–246.
[30] J.E. Van Aken, Management research as a design science: articulating the research products of mode 2 knowledge production in management, British Journal of Management 16 (1) (2005) 19–36.
[31] R. Wieringa, A. Moral, Technical action research as a validation method in information systems design science, International Conference on Design Science Research in Information Systems, Springer, 2012.
[32] Alarcos Research Group, ArchiRev Tool, [cited 2019 28/08/2019]; Available from, 2019. http://infalarcosj.esi.uclm.es/ArchiRev/.
[33] F. Wilhelmstötter, Jenetics. Genetic Programming Library, 20/08/2019 [cited 2019 20/08/2019]; Available from, 2019. http://jenetics.io/.
[34] H. Reefke, D. Sundaram, Sustainable supply chain management: decision models for transformation and maturity, Decision Support Systems 113 (2018) 56–72.
[35] R. Perez-Castillo, ArchiRev-VS-script, [cited 2019 13/05/2019]; Available from, 2019. https://alarcos.esi.uclm.es/per/rpdelcastillo/Ex_EA_ArchiRev/archirev-vs-script.rar.
[36] F. Gomez, A. Quesada, Genetic algorithms for feature selection. Machine learning blog, [21/08/2019]; Available from, 2019. https://www.neuraldesigner.com/blog/genetic_algorithms_for_feature_selection.
[37] K. Kalaiselvi, A. Kumar, An empirical study on effect of variations in the population size and generations of genetic algorithms in cryptography, 2017 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC), 2017.
[38] O. Roeva, S. Fidanova, M. Paprzycki, Influence of the population size on the genetic algorithm performance in case of cultivation process modelling, 2013 Federated Conference on Computer Science and Information Systems, 2013.
[39] S.N. Sivanandam, S.N. Deepa, Introduction to Genetic Algorithms, 1 ed., vol. XIX, Springer-Verlag, Berlin Heidelberg, 2008, p. 442.
[40] P.K. Chawdhry, R. Roy, R.K. Pant, Soft Computing in Engineering Design and Manufacturing, Springer Science & Business Media, 2012.

Ricardo Pérez-Castillo is a researcher at the Information Technologies and Systems Institute, University of Castilla-La Mancha (UCLM), Spain. His research interests include architecture-driven modernization, model-driven development, business-process archaeology, and enterprise architecture. Perez-Castillo received a Ph.D. in computer science from UCLM. Contact him at ricardo.pdelcastillo@uclm.es



Francisco Ruiz is a full professor at the Information Technologies and Systems Institute, University of Castilla-La Mancha (UCLM), Spain. His research interests include enterprise architecture, business-process technology, and software engineering. Ruiz received a Ph.D. in computer science from UCLM. Contact him at francisco.ruizg@uclm.es

Mario Piattini is the director of the Alarcos Research Group and a full professor at the University of Castilla-La Mancha, Spain. His research interests include software and data quality, information-systems audit and security, and IT governance. Piattini received a Ph.D. in computer science from Madrid Technical University, Spain. Contact him at mario.piattini@uclm.es