Journal of Computer Science

Original Research Paper

An Algorithm To Determine The Maturity Improvement Plan For Information System Risk Management. Application On A Case Study

[1]Soumaya Amraoui, [1,2,3]Mina Elmaallam and [3]Hicham Bensaid

[1]IMS Team, ADMIR Laboratory, Rabat IT Center, ENSIAS, Mohammed V University, Rabat, Morocco
[2]LYRICA Laboratory, School of Information Sciences, Rabat, Morocco
[3]Smart, Embedded, Enterprise and Distributed Systems (SEEDS) CEDOC-2TI, INPT, Rabat, Morocco

Article history:
Received: 27-03-2019
Revised: 20-05-2019
Accepted: 31-07-2019

Corresponding Author:
Soumaya Amraoui
IMS Team, ADMIR Laboratory, Rabat IT center, ENSIAS, Mohammed V University, Rabat, Morocco
Email: soumaya.amraoui@gmail.com

Abstract: A good and relevant Risk Management process is a key issue when Information System effective governance is concerned. Therefore, several paradigms have been devised to help achieving such goal. Among these paradigms, maturity models are quite popular. The main aim of a maturity model is to help users improve their activities capability. However, one of the major challenges encountered when using these models is the definition of the improvement plan after the evaluation. This challenge is all the stronger and costly when it comes to an activity whose elements or phases have an important interdependence such as IS risk management. In this article, we propose an algorithm called "Path Prerequisites" to help users define a graduate improvement plan, easily and efficiently, from a given maturity level to a target one, while handling criteria dependencies constraints. The algorithm is based on an acyclic graph representation of the control objectives and the dependencies among them and it corresponds to a guided (backwards) traversal of the graph. We assess the algorithm by applying it to a study case.

Keywords: Information System, Maturity, Maturity Model, Focus Area Model, Risk Management

Introduction

The governance of an information system is the definition and the implementation of the strategy and the necessary tools for the achievement of its objectives. However, these objectives can be achieved only if the information system is protected against any potential threats through the implementation of an effective Risk Management (RM) process. Hence the need of a maturity model for information system risk management process.

A Maturity Model is a technique that has been proved to be valuable in measuring different aspects of a process or an organization (Proenca and Borbinha, 2016). It identifies deficiencies in process structure and management and unsatisfactory performance causes (Mayer and Fagundes, 2009). Maturity models are available to respond to many different challenges (Carvalho *et al*., 2019). They provide information for organizations to address the problems and challenges in a structured way, providing both a reference point to assess the capabilities and a road map for improvement (Caralli and Knight, 2012b). Based on the assumption of predictable patterns of organizational evolution and change, maturity models typically represent theories about how an organization's capabilities evolve in a stage-by-stage manner along an anticipated, desired, or logical path. They give guidance through an evolutionary process by incorporating formality into the promizing improvement activities (Mettler, 2010). These tools allow self-assessment and provide a relevant benchmark of these activities in relation to best practices (Elmaallam and Kriouile, 2013). During the last five decades, several maturity models have been proposed, which differ not only in terms of the number of stages, but also on maturity-influencing factors and intervention fields (Rocha, 2011). However, to be effective a maturity model should be relevant and should deal with the real issue of the targeted assessment. The first question that organizations and researchers try to answer is then: which maturity model best meets the requirements of the assessed activity? The second and more important question is: what efficient method to use for defining the improvement plan after assessing an activity? This question is more important when there is

interdependency between the assessed activity elements which involves significant effort to elaborate improvement plan. To the best of our knowledge, no existing work adequately addresses this last issue. They rather only treat the assessment aspect. In this article we focus on the Focus Area (FA) model and propose an algorithm ("Path Prerequisites") to assist its users to define an improvement plan once the evaluation of the maturity is made. The remainder of this paper is structured as follows: Section two presents the research methodology. Section three gives the research background as a prerequisite for the "design and development" phase. Section three introduces the "Path Prerequisites" algorithm. Section four illustrates the "demonstration and evaluation" phases which are an application of the "Path Prerequisites" algorithm to the model "ISR3M" (Elmaallam *et al*., 2019), a maturity model for assessing the information system risk management. Section five concludes.

Research Methodology

We start by introducing the Design Science Process that we use for presenting our research approach (Peffers *et al*., 2007). The Design Science Research (DSR) methodology (Hevner *et al*., 2004) focuses on the development of a new artefact. It is particularly suitable for research on the process assessment discipline (Carlsson *et al*., 2011). In order to develop our artefact, we follow the six phases of the DSR approach:

1. **Phase 1: Problem identification and motivation**
   The problem motivating our work is how to develop an approach for defining maturity improvement plan while optimizing efforts for information system risk management
2. **Phase 2: Define the objectives for a solution**
   The objective of our solution is to devise an algorithm, called "Path Prerequisites", to define an improvement plan for IS risk management maturity model
3. **Phase 3: Design and development**
   The "Path Prerequisites" algorithm Focus Area maturity model development method is based on the Focus Area maturity matrix, as described in background section and graph analysis method to compute all possible paths from a target configuration to an initial configuration while optimizing the effort
4. **Phase 4 and 5: Demonstration and Evaluation**
   For these two phases, an application of the algorithm is presented in the fifth and sixth sections. It concerns the definition of improvement plan of information system risk management maturity model
5. **Phase 6: Communication**
   This phase is assured by publication of the algorithm in scientific conferences and journals and its intended use by professionals, especially in information system risk management maturity improvement

Background

Information System as a Work System

Several definitions of an Information System (IS) exist in the literature (Carvalho *et al*., 2019). In our work, we adopt that of the IS as a Work System (WS) (Alter and Sherer, 2004). We opt for this definition since it clearly identifies the components of an IS and thus provide more relevant results of IS risk management activities. A *Work System* is a system (as depicted on Fig. 1) in which human participants and/or machines perform work (processes and activities) using the information, technology and other resources to produce specific products and/or services for of internal or external customers (Alter and Sherer, 2004).

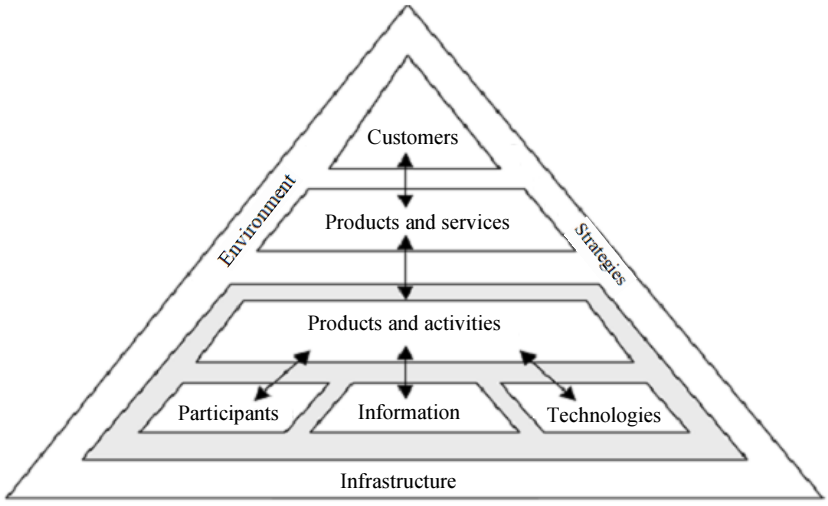

Fig. 1: The work system framework (Alter and Sherer, 2004)

An information system is a work system whose processes and activities are devoted to processing information, that is, capturing, transmitting, storing, retrieving, manipulating and displaying information (Alter and Sherer, 2004).

IS Risk Management

Risk management can be defined as "the set of coordinated activities for the purpose of leading and controlling an organization toward risk" (ISO, 2009b). Moreover, IS risk management is an ongoing business of identifying and mitigating risks (Alter and Sherer, 2004). IS risk management activities must be included in a realistic model that describes efficiently the overall system of the organization, because the recognition of risk factors encourages appropriate risk reduction tactics (Alter and Sherer, 2004). For example, project managers who don't have an essential skill can use an employee or consultant having such ability or completely modify the project skills (Alter and Sherer, 2004) otherwise. The risk reduction tactics available depend on the objectives and expectations that apply (Alter and Sherer, 2004). For example, a project that aims to minimize costs finds additional difficulty in hiring expensive consultants (Alter and Sherer, 2004). Radut (2009) considers that IS risk management is "a framework for classifying, assessing and mitigating IS risks until achieving an acceptable threshold". Olzak (2008) argues that information risk management is "the proper employment of tools and methods leading to security controls implementation allowing a business risk extenuation and by then insuring information performance. This must be done in a way that maintains for each personnel and processes, protected by these controls and using the systems, to the highest level of prospective operational efficiency." When developing its IS security risk model, Mayer and Fagundes (2009) adopts the definition of ISO (2009a), which considers that IS risk management is "the set of coordinated activities to guide and control an organization in relation to the IS risks to which it is exposed". According to Valentin and Vasile (2008), IT risk management consists of analyzing the risk knowledge taken by the company through its IT systems in terms of business impact. Salvati (2008) believes that beyond the procedural aspects that are often emphasized in the description of IS risk management activities, it is necessary to highlight the decision-making aspects as well. In such an interpretation, IS risk management represents a structured approach to risk-informed decision-making which aligns the functioning of the enterprise information system to its risk appetite (Salvati, 2008). In the same vein, the Risk Management Guide for US Department of Commerce Information Technology Systems argues that information risk management must exist not only to protect its IT assets but also to "protect organization and its ability to fulfill its mission", (Woodall *et al*., 2014). Therefore, the risk management process should not be primarily treated as a technical function performed by IT experts who operate and manage the computer system, but as a core management function of the organization (Stoneburner *et al*., 2002). Information risk management needs to be incorporated into all decisions and everyday operations, can afterwards provided to be used effectively, regarded as tool to manage information proactively rather than reactively. Managing the risks of an information system involves managing the risks of its nine axes in relation to their evaluation elements. According to Elmaallam and Kriouile (2015), the evaluation elements of each axis are identified through (1) the missions and requirements of the work system framework as defined in the literature (Alter and Sherer, 2004), (2) the application of the theory Resource Based-View (RBV) (Wade and Hulland, 2004) on IS defined as WS considering both dynamic resources such as skills, as static as the technical infrastructure, (3) the IS risk factors (Alter and Sherer, 2004) and (4) interviews with IS experts. Table 1 lists the evaluation elements for each component.

Maturity Model Architectures

Maturity models typically include a sequence of levels (or stages) that form an anticipated, desired and logical path from an initial state to maturity (Röglinger *et al*., 2012). An organization's current maturity level represents its capabilities regarding specific class of objects and application domain (Rosemann and de Bruin, 2005). Maturity models are used to assess as-is situations, to guide improvement initiatives and to control progress (Iversen *et al*., 1999). After defining the maturity level of an activity or process, users have to define an improvement plan. The latter corresponds to the set of actions that must be achieved to reach a desired level maturity of the assessed activity. There are three types of maturity model architectures (Van Steenbergen *et al*., 2007). The first two architectures are qualified as "Fixed Level Architectures". These are "staged" and "continuous" architectures. The staged architecture is characterized by several Maturity Levels (ML). Every level groups a set of maturity domains. A level is reached if all requirements of its domains are verified. Table 2 illustrates the staged architecture which has n levels. Domain k having the level n means that all requirements of this domain for this level are verified. The organization can have different levels for different domains.

The continuous architecture measures the domain's capacity. It defines a scale of skill levels for the latter. A domain reaches a level of aptitude if it satisfies all the corresponding requirements.

Table 3 illustrates the continuous architecture. An organization having the level *j* means that all corresponding domains verify the requirement of this level. In the same example, the activity has level 2. The most recognized model in this architecture is Capability Maturity Model Integration (CMMI). The later addresses three areas of interest: Product and service development, Service establishment and management and finally Product and service acquisition. It has level 5 for the both staged and continuous architectures. The CMMI-Like models are the models which use the CMMI architecture but for other disciplines. They are widely used but present certain limits. The most important limit, in the present research context, is the strong focus on formalization of improvement activities accompanied by extensive bureaucracy (Herbsleb and Goldenson, 1996), in absence of formal method which can help in fast, not expensive and reliable decision-making. The third type is the test process improvement model proposed by (Koomen and Baarda, 2006). This is the "Focus Area model" (FA). It is based on the idea that each area of maturity has its own evolution. It is interesting for assessing activity with interdependencies between their various domains. For this reason, the FA is the most adequate model for risk management process and is then developed in section 3.4.

Table 4 illustrates this architecture. The organization has level 2. But each domain has its own level. Domain *1* has level *m*. domain 2 has level 2. Etc. The FA model is detailed in section 3.4.

Table 1: Focus Area architecture

Axis	Evaluation element
IT	Complexity
	Modifiability
	Competitive importance
	Potential of credibility
	Strategic profile
Participants	Competence
	Cooperation
	Stability
Information	Security
	Reliability
	Relevance
Process	Agility
	Formalization
	Updating
	Interaction
	Coherence
Product	Compliance with requirements
	Quality
	Exploitation
Customer	Needs
	Satisfaction
	Competence
	Cooperation
Infrastructure	Technical infrastructure
	Human infrastructure
	Informational infrastructure
Strategy	Alignment
	Contribution
Environment	Culture
	Intra Enterprise regulations

Table 2: Fixed level (*n*) staged architecture

	Level 1	Level 2	…	Level *n*
Domain 1	X			
Domain 2	X			
Domain 3		X		
…				
Domain *k*				X

Table 3: Fixed Level (*j*) continuous architecture

	Level 1	Level 2	…	Level *j*
Domain 1	X	X	X	X
Domain 2	X	X		X
Domain 3	X	X	X	
…	X	X	X	
Domain *k*	X	X	X	X

Table 4: Focus area architecture

	Level 1	Level 2	…	Level *m*	…
Domain 1	X	X	X	X	
Domain 2		X			
Domain 3		X		X	X
…	X	X	X		
Domain *k*		X	X		X

Focus Area Model

"Focus Area (FA)" (Steenbergen *et al*., 2010) is a maturity model design approach developed using the Design Science Research (DSR) process (Peffers *et al*., 2007). FA Maturity models aim to support the continuous and progressive improvement of software testing (Koomen and Baarda, 2006). A Focus Area is a well-defined coherent subset of a Functional Domain (Steenbergen *et al*., 2010). The total set of focus areas is a partition of the functional domain, i.e. different focus areas are disjointed and the union of all these focus areas is the complete functional domain (Steenbergen *et al*., 2010). In this category of models each focus area has its own number of specific maturity levels. The overall maturity of an organization is expressed as a combination of the maturity levels of these focus areas. The approach proposed by (Steenbergen *et al*., 2010) consists of four steps: (1) Scoping: Identify and scope domain, (2) design model: Determine focus area, capabilities, dependencies and position capabilities in matrix, (3) Instrument development: Develop assessment instrument and define improvement actions, (4) implementation and exploitation: Implement maturity model, improve matrix iteratively and communicate results.

The proposed approach illustrated in Fig. 2 is modeled using the notation presented by (Weerd and Brinkkemper, 2008), which is based on standard UML conventions, with some minor adjustments. The maturity matrix is the key deliverable of the design phase. It includes FA capabilities (or Control Objectives (CO): A, B, C, etc.) which give a score for each activity domain or

Focus Area. Those capabilities are based on their order and dependencies. It provides the maturity level once the instrument has been designed and also defines improvement paths. An organization reaches overall maturity level '*l*' (0≤*l*≤ *max* levels defined in matrices) if:

- All capacities located in the column corresponding to the level '*l*' are verified
- All capacities located in the left of the column corresponding to the level '*l*' are verified
- There is at least one capacity on the right of the column corresponding to the level '*l*' that is unverified

Figure 3 gives an example of an FA maturity matrix. This later contains 18 domains and 13 levels. The first "Development of Architecture (DA)" has three control objectives: DA.A, DA.B and DA.C. The second domain "Use of Architecture (UA)" has three control objectives: UA.A, UA.B and UA.C. The third domain "Alignment with Business (AB)" has also three control objectives: AB.A, AB.B, AB.C. Figure 3 illustrates the interdependencies between the control objectives of the two domains.

For example, according to the interdependencies Table 3, DA.C depends on DA.B which depends on DA.A. This means that this control objective cannot be achieved unless the two others control objectives are achieved. The symbol *X* means a direct dependence. The symbol (*X*) means a transitive dependence.

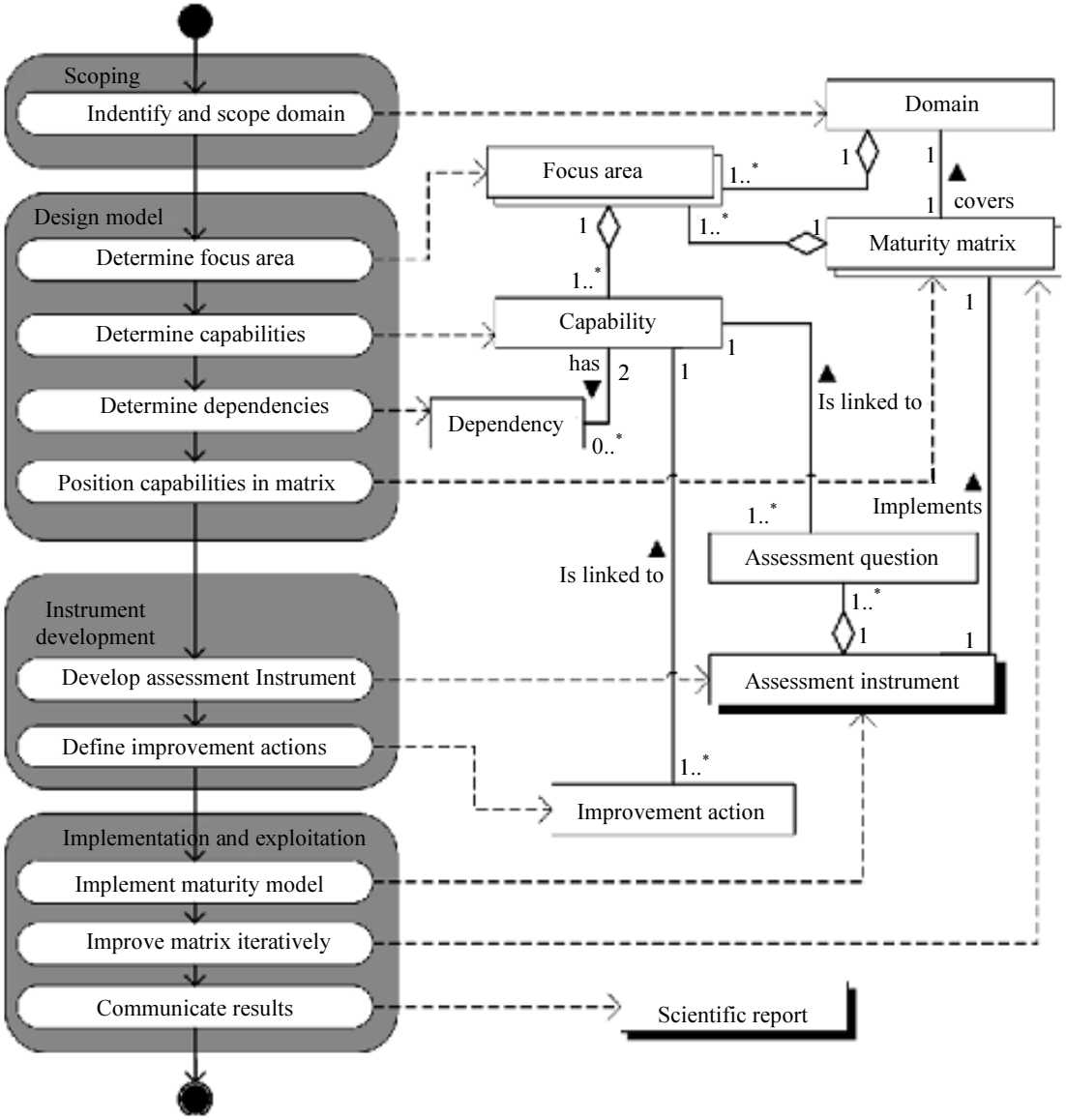

Fig. 2: The development method for focus area maturity models (Steenbergen *et al*., 2010)

Focus area \ Maturity scale	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Development of architecture		A			B			C						
Use of architecture			A			B				C				
Alignment with business		A				B				C				
Alignment with development process			A				B		C					
Alignment with operations					A			B			C			
Relationship to the as-is state					A				B					
Roles and responsibilities				A		B					C			
Coordination of developments							A			B				
Monitoring				A		B		C		D				
Quality management								A		B			C	
Maintenance of the architectural process							A		B		C			
Maintenance of the architectural deliverables					A			B					C	
Commitment and motivation		A					B		C					
Architectural roles and training				A		B			C			D		
Use of an architectural method				A						B				C
Consultation			A		B				C					
Architectural tools							A				B			C
Budgeting and planning				A							B		C	

Fig. 3: Example of FA model maturity matrix

Related Work

Related work on the literature can be classified into three axes: (1) Generics risk management maturity models, (2) Generics project risk maturity models and (3) Information System (or one of its components like IT) risk management maturity models. The comparative analysis of maturity models aims to evaluate them based on a set of well-defined criteria. These are deduced from the problem as well as the background elements. These criteria are:

- Genericity: The proposed solution must be generic from the point of view of the IS risk management process and concept
- Independence of the context of application: The solution must be applicable in all the contexts and sectors of activity
- Adaptability: The solution must make it possible to take into account the specificities of the context where the model is applied
- Transparency: The solution must ensure the documentation and the traceability of the measures of maturity
- Improvement plan: Does the model assist its users in defining an improvement plan?
- Theoretical basis: Is the model based on the theoretical aspect of the domain studied for measuring maturity?
- Suitability for needs (IS RM): Is the model suitable for IS risk management and consider all aspects of an IS?

Table 5 presents the result of the evaluation of the maturity models according to the criteria mentioned above.

The results show that the model that best meets the evaluation criteria is the ISR3M model. The second observation is that none of the studied models presents an accompaniment to assist users in the development of the improvement plan resulting from the evaluation of the maturity.

Next section presents our proposed algorithm to define an improvement plan for the ISR3M model and all model based on FA architecture which is the best suitable model for risk management maturity model evaluation.

Improvement Plan Definition Algorithm: "Path Prerequisites"

The aim of the focus area maturity model is to describe a functional domain using a partition of adequate criteria which allows a more accurate assessment of the maturity. In particular, each criterion is given a score ranging over A, B, C, … depending on the maturity of the criterion. The set of (partial) scores gives an overall maturity score for the functional domain. Moreover, focus area criteria are defined using dependency (binary and transitive) relations stating for instance that some criterion cannot have the score v unless some other criteria have already reached some values (defined by the model). For example, we can have in a model that the focus area (or criterion) X cannot reach the value $X.B$ unless some criteria Y and Z have respectively the values $Y.B$ and $Z.C$. Hence, in order to define a clear improvement plan to go from an Initial

Configuration (IC) to a Target one (TC), we have to consider dependency relations and propose intermediate improvement leading (among others) to reach (if it is not already the case) the scores *Y.B* and *Z.C*. Such a task can be tricky especially if the dependency relations are somewhat complex. To the best of our knowledge, no work has been done to tackle this aspect, i.e., no algorithm has been considered to propose, given a focus area maturity model and a set of dependencies, an improvement plan allowing to reach a target configuration TC from an initial configuration IC. In the following, we propose such an algorithm. First, we model the dependency rules using a directed acyclic graph where the set of vertices corresponds to the set of all possible scores for all focus areas. Hence if we consider for example a focus area X which can have three score values A, B and C, we shall then consider three vertices *X.A*, *X.B* and *X.C*. The edges of the graph are based on direct dependencies. If a focus area X depends on a focus area *Y* such that for *X* to reach the value v_X it is mandatory that *Y* reaches the value v_Y then in the graph we will have a directed edge $e = (X.v_X, Y.v_Y)$. The resulting graph describes the interdependencies between focus area values. Formally, we define the focus area interdependencies graph FAIG as:

FAIG = (*S*,*R*) where *S* is a set of vertices such that: $S = \cup S_F A$ and $S_{FA} = \cup_i \{[\![FA]\!]_i\}$ with each $[\![FA]\!]_i$ stands for a different score value of FA.

R is the set of arcs in the graph, $R = \{(X.v_X, Y.v_Y)\}$ such that $X.v_X, Y.v_Y \in S$ and it is not possible to reach the score value vX for the focus area *X* if the score value v_Y of *Y* is not yet reached.

Actually, a score value v_X for a focus area *X* can depend on more than one other focus area *Y*. We define R^* the transitive closure of *R* (seen as a binary relation).

We propose an algorithm to compute all possible paths from a target configuration (the set of all scores we want) to an initial configuration (the set of all current scores). We give an example (Fig. 5) based on the interdependencies matrix above (Fig. 4) to illustrate our aim: This graph specifies that *DA.B* cannot be reached if *DA.A* is not, that *UA.B* cannot be reached if both *DA.B* and *UA.A* are not and so on.

If we consider that initially we have the set {*DA.A*,*UA.B*,*AB.A*} and we want to reach the target configuration {*DA.B*,*UA.C*,*AB.C*} then we have to reach respectively, *DA.B*,*UA.B* and *AB.B*. We will have actually the paths depicted on Fig. 6.

In our setting, the dependencies are considered as conjunctive ones. For example, considering the dependency graph of Graph 1, *AB.C* cannot be reached unless both *UA.B* and *AB.B* are reached. Improvement plans we need an exhaustive traversal of the graph going from the target vertices to current ones. Since the graph is acyclic, the algorithm always converges. The algorithm is straightforward, actually, we start from the target configuration (a set of target Control Objectives (COs) and we follow backwards direct connections in the graph to reach the initial configuration (the curret set of COs). As said before, since the graph is guarateed to be acyclic, our (backwards) graph traversal converges. Algorithm 1 provides a full description of the "Path Prerequisites" algorithm.

Depends on	DA.A	DA.B	DA.C	UA.A	UA.B	UA.C	AB.A	AB.B	AB.C
DA.A									
DA.B	X								
DA.C	(X)	X							
UA.A	X	--	--						
UA.B	(X)	X	--	X					
UA.C	(X)	(X)	X	(X)	X				
AB.A	--	--	--	--	--	--			
AB.B	(X)	--	--	X	--	--	X		
AB.C	(X)	(X)	--	(X)	X	--	(X)	X	

Fig. 4: Example of interdependencies matrix

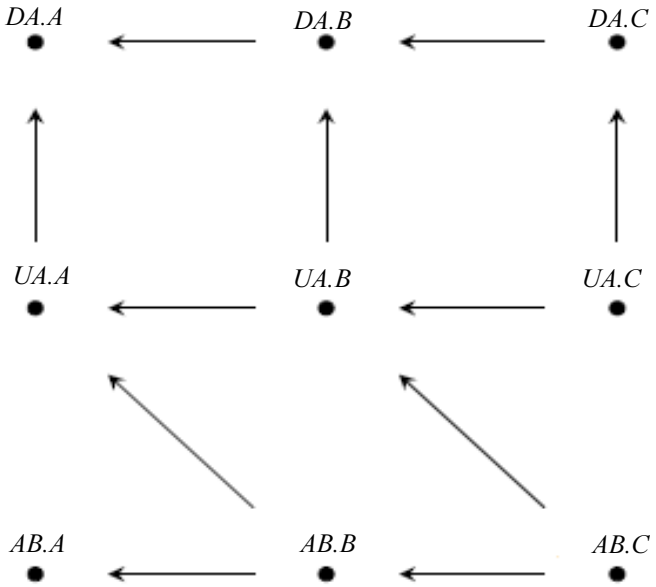

Fig. 5: FAIG example

$$DA.C \rightarrow DA.B \rightarrow DA.A$$
$$UA.C \rightarrow UA.B \rightarrow DA.B \rightarrow DA.A$$
$$UA.C \rightarrow UA.B \rightarrow UA.A$$
$$AB.C \rightarrow AB.B \rightarrow AB.A$$
$$AB.C \rightarrow AB.B \rightarrow UA.A$$
$$AB.C \rightarrow UA.B \rightarrow DA.B \rightarrow DA.A$$
$$AB.C \rightarrow UA.B \rightarrow UA.A$$

Fig. 6: Resulting paths

Table 5: Comparative analysis of maturity models

Model	Objective	Genericity	Independence of the context of application	Adaptability	Transparency	Improvement plan	Theoretical basis	Suitability for needs (IS RM)
RMM	Risk maturity Model (Hillson, 1997)	-	+	-	-	-	-	-‘
Project RMM	Project Risk Maturity Model (Hopkinson, 2011)	-	+	-	+	-	-	-
COPS	Risk Management Capability Maturity Model for Complex Product Systems Projects (Ren and Yeo, 2004)	-	+	-	+	-	-	-
ERMM Level Assessment Tool	Enterprise Risk Management Maturity-Level Assessment Tool (Caralli and Knight, 2012a)	+	+	-	+	-	-	-
IT Risk Management: A Capability Maturity Model	IT Risk Management (Carcary, 2013)	-	+	-	+	-		+ (IT)
CMMI	Guide for improving software development and maintenance practices. Guide for improving software development and maintenance practices (Basque, 2011)	-	+	-	+	-	+	+ (Software development)
MMGRSeg	Measures the maturity level of the information security related to risk management process (Mayer and Lemes Fagundes, 2009)	+	+	-	-	-	-	+ (IT security)
IT Risk maturity model	Framework proposed by the Information Systems Audit and Control Association (ISACA) for IT risk management (Isa, 2012)	-	+	-	+	-	+	+
ISR3M	Information System Risk Management Maturity Model (Elmaallam *et al*., 2019)	+	+	+	+	-	+	+

Application of the "Path Prerequisites" Algorithm on ISR3M Maturity Model

Information Systems Risk Management Maturity Model (ISR3M) (Elmaallam *et al*., 2019) aims to evaluate Information System (IS) risk management. An IS is defined as a special case of work system (Alter and Sherer, 2004). The elements to consider in the study of such systems are: participants, information, technology, processes, products and services, customers, infrastructure, environment and strategy. As for the risk management, we adopt the ISO 31000 Framework (ISO, 2009a) with the generic management cycle proposed by Sienou (2009). This cycle resumes the stages of the process proposed by ISO 31000 with a restructuring of its phases. Indeed: (1) Communication is considered as an activity inherent to every phase of the process (Sienou, 2009), (2) the cycle of management preserves its iterative character, but no longer requires synchronization of all stages with a monitoring phase (Sienou, 2009) and (3) Treatment may be the cause of a new iteration process (Sienou, 2009). The development of ISR3M model should provide answers to the problem of assessing IS risk management from two perspectives (Elmaallam and Kriouile, 2015). The first perspective is academic. The model must address a problem not sufficiently addressed in IS research: The assessment of IS risk management. The proposed solution must also be able to open new perspectives and opportunities in scientific research in this area. The second perspective relates to the practical side. The proposed model should be easy to implement and comply with the best practices of risk management. This model is developed using MMDPIS process (Elmaallam and Kriouile, 2014) and aims to satisfy seven principal requirements: (1) Genericity, (2) Independence of application context, (3) adaptability, (4) transparency, (5): plan improvement, (6) Theoretical basis and (7) Need adequacy (IS RM topic) (Elmaallam and Kriouile, 2015). It's structured along two dimensions. The first dimension includes evaluated activities. It is a matter of risk management activities.

N°	Area	0	1	2	3	4	5	6	7	8	9	10	11	12
1	**RM Principles (PRM)**		A	B	C									
	Organizational Framework													
2	Mandate and commitment (ME)		A	B										
3	Framework design (CCO)			A	B	C	D	E						
4	Risk management implementation (MOE)					A		B	C					
5	Monitoring and review (SRC)						A			B	C			
6	Continual improvement (ACC)							A			B	C		
	Process													
	Establishment of context													
7	External context (ECX)		A	B	C									
8	Internal context (ECP)		A	B	C									
9	Process context (ECP)		A		B	C	D							
10	Risk management criteria (ECC)			A			B	C						
	Risk assessment													
11	Risk identification (API)				A		B	C	D	E				
12	Risk analysis (APA)					A		B	C					
13	Risk evaluation (APV)					A			B	C				
	Treatment													
14	Selection of treatment option (TSO)					A				B	C			
15	Elaboration of treatment plan (TEP)					A					B	C		
16	Implementing of treatment plan (TMP)						A					B	C	
17	Process monitoring and review (SR)							A					B	C
18	Recording (Eng)		A	B	C									

Fig. 7: ISR3M positioning maturity matrix (Elmaallam *et al*., 2019)

N°	Area	0	1	2	3	4	5	6	7	8	9	10	11	12
1	RM principles (PRM)		A	B	C									
2	Mandate and commitment (ME)		A	B										
3	Design of framework (CCO)			A	B	C	D	E						
4	Risk management implementation (MOE)					A		B	C					
5	Monitoring and review (SRC)						A			B	C			
6	Continual improvement (ACC)							A			B	C		
7	External context (ECX)		A	B	C									
8	Internal context (ECP)		A	B	C									
9	Process context (ECP)		A		B	C	D							
10	Risk management criteria (ECC)			A			B	C						
11	Risk identification (API)				A		B	C	D	E				
12	Risk analysis (APA)					A		B	C					
13	Risk evaluation (APV)					A			B	C				
14	Selection of treatment option (TSO)					A				B	C			
15	Elaboration of treatment plan (TEP)					A					B	C		
16	Implementing of treatment plan (TMP)						A					B	C	
17	Monitoring and review (SR)							A					B	C
18	Recording (Eng)		A	B	C									

Fig. 8: Example of maturity matrix for initial configuration

N°	Area	0	1	2	3	4	5	6	7	8	9	10	11	12
1	RM principles (PRM)		A	B	C									
2	Mandate and commitment (ME)		A	B										
3	Design of framework (CCO)			A	B	C	D	E						
4	Risk management implementation (MOE)					A		B	C					
5	Monitoring and review (SRC)						A			B	C			
6	Continual improvement (ACC)							A			B	C		
7	External context (ECX)		A	B	C									
8	Internal context (ECP)		A	B	C									
9	Process context (ECP)		A		B	C	D							
10	Risk management criteria (ECC)			A			B	C						
11	Risk identification (API)				A		B	C	D	E				
12	Risk analysis (APA)					A		B	C					
13	Risk evaluation (APV)					A			B	C				
14	Selection of treatment option (TSO)					A				B	C			
15	Elaboration of treatment plan (TEP)					A					B	C		
16	Implementing of treatment plan (TMP)						A					B	C	
17	Monitoring and review (SR)							A					B	C
18	Recording (Eng)		A	B	C									

Fig. 9: Example of maturity matrix for target configuration

The projected improvements are illustrated in Table 6. Areas which are not subject of improvement are not mentioned

Results

In this section, we present the exhaustive result of applying "Path Prerequisites" algorithm on ISR3M maturity model.

Table 7 provides all required CO to achieve the target set given in Table 6.

According to the results above, the treatment plan should contain the actions required to achieve the control objectives of each area. For example, to achieve the APA.B, the organization must:

- Use tools and techniques for risks analysis of all IS component
- Ensure obtaining results of analysis allowing estimating the risks
- Define the strategies and the methods of treatment during the analysis
- Consider and communicate to the stakeholders the degrees of confidence in the determination of the level of the risk and its sensibility in prerequisites and in hypotheses
- Mention and underline factors such as: the difference of opinion between experts (IT, process), the uncertainty, the availability, the quality, the quantity and the validity of the relevance of the data/ information or the limits of modellings

Table 6: Target control objectives by area

	CO initial Set	CO Target set
PRM	A	B
ME	A	B
CCO	A	B
MOE	--	B
SRC	--	B
ACC	--	A
ECI	B	C
ECP	A	D
ECC	--	C
API	A	E
APA	A	C
APV	A	C
TSO	A	C
TEP	A	C
TMP	A	C
SR	--	C
Eng	--	C

Table 7: Control objective to achieve

Area	Control objective
RM principles (PRM)	PRM.B
Mandate and commitment (ME)	ME.B
Framework design (CCO)	CCO.B, CCO.C
Risk management implementation (MOE)	MOE.A, MOE.B
Monitoring and review (SRC)	SRC.A, SRC.B
Continual improvement (ACC)	ACC.A
External context (ECX)	--
Internal context (ECI)	ECI.C
Process context (ECP)	ECP.B, ECP.C, ECP.D
Risk Management criteria (ECC)	ECC.A, ECC.B; ECC.C
Risk Identification (API)	API.B, API.C, API.D, API.E
Risk Analysis (APA)	APA.B, APA.C
Risk Evaluation (APV)	APV.B, APV.C
Selection of treatment Option (TSO)	TSO.B, TSO.C
Elaboration of treatment plan (TEP)	TEP.B, TEP.C
Implementing treatment plan (TMP)	TMP.B, TMP.C
Monitoring and review (SR)	SR.A; SR.B, SR.C
Recording (Eng)	Eng.A, Eng.B, Eng.C

Table 8: "Path Prerequisites" results for RM principles

Area	Init. CO	Targ. CO	Ways
PMR	A	B	pmr.b->pmr.a->

Table 9: "Path Prerequisites" results for organizational framework

Area	Init. CO	Targ. CO	Ways
ME	A	B	me.b->me.a->
CCO	A	B	cco.b->me.a-> cco.b->me.b->me.a->
MOE	--	B	moe.b->moe.a->cco.a-> moe.b->cco.c->cco.b->me.a moe.b->cco.c->cco.b->me.b->me.a->
ECC	--	C	src.b->src.a->moe.a->cco.a-> src.b->moe.b->moe.a->cco.a src.b->moe.b->cco.c->cco.b->me.a-> src.b->moe.b->cco.c->cco.b->me.b->me.a->
ACC	--	A	acc.a->src.a->moe.a->cco.a->

Table 10: "Path Prerequisites" results for context establishment activity of RM process

Area	Init. CO	Targ. CO	Ways
ECX	C	C	---
ECI	B	C	eci.c->eci.b->
ECP	A	D	ecp.d->ecp.c->ecp.b->
ECC	--	C	ecc.c->ecc.b->ecp.b->ecc.a-> ecc.c->ecp.c->ecp.b->

Table 11: "Path Prerequisites" results for appreciation activity of RM process

Area	Init. CO	Targ. CO	Ways
API	A	E	api.e->api.d->api.c->api.b->api.a-> api.e->api.d->api.c->api.b->ecp.c->ecp.b
APA	A	C	apa.c->apa.b->apa.a-> apa.c->apa.b->ecp.c->ecp.b-> apa.c->api.b->api.a-> apa.c->api.b->ecp.c->ecp.b->
APV	A	C	apv.c->apv.b->apv.a-> apv.c->apv.b->apa.b->apa.a-> apv.c->apv.b->apa.b->ecp.c->ecp.b-> apv.c->apa.c->apa.b->apa.a-> apv.c->apa.c->apa.b->ecp.c->ecp.b-> apv.c->apa.c->api.b->api.a-> apv.c->apa.c->api.b->ecp.c->ecp.b->

Table 12: "Path Prerequisites" results for monitoring and review activity of RM process

Area	Init. CO	Targ. CO	Ways
SR	--	C	sr.c->sr.b->sr.a->tmp.a->

Table 13: "Path Prerequisites" results for monitoring and review activity of RM process

Area	Init. CO	Targ. CO	Ways
Eng	--	C	eng.c->eng.b->eng.a->

Table 14: "Path Prerequisites" results for treatment activity of RM process

Area	Init. CO	Targ. CO	Ways
TSO	A	C	tso.c->tso.b->tso.a-> tso.c->tso.b->apv.b->apv.a-> tso.c->tso.b->apv.b->apa.b->apa.a-> tso.c->tso.b->apv.b->apa.b->ecp.c->ecp.b-> tso.c->apv.c->apv.b->apv.a-> tso.c->apv.c->apv.b->apa.b->apa.a-> tso.c->apv.c->apv.b->apa.b->ecp.c->ecp.b-> tso.c->apv.c->apa.c->apa.b->apa.a-> tso.c->apv.c->apa.c->apa.b->ecp.c->ecp.b-> tso.c->apv.c->apa.c->api.b->api.a-> tso.c->apv.c->apa.c->api.b->ecp.c->ecp.b->
TEP	A	C	tep.c->tep.b->tep.a-> tep.c->tep.b->tso.b->tso.a-> tep.c->tep.b->tso.b->apv.b->apv.a-> tep.c->tep.b->tso.b->apv.b->apa.b->apa.a-> tep.c->tep.b->tso.b->apv.b->apa.b->ecp.c->ecp.b-> tep.c->tso.c->tso.b->tso.a-> tep.c->tso.c->tso.b->apv.b->apv.a-> tep.c->tso.c->tso.b->apv.b->apa.b->apa.a-> tep.c->tso.c->tso.b->apv.b->apa.b->ecp.c->ecp.b-> tep.c->tso.c->apv.c->apv.b->apv.a-> tep.c->tso.c->apv.c->apv.b->apa.b->apa.a-> tep.c->tso.c->apv.c->apv.b->apa.b->ecp.c->ecp.b-> tep.c->tso.c->apv.c->apa.c->apa.b->apa.a-> tep.c->tso.c->apv.c->apa.c->apa.b->ecp.c->ecp.b-> tep.c->tso.c->apv.c->apa.c->api.b->api.a-> tep.c->tso.c->apv.c->apa.c->api.b->ecp.c->ecp.b->
TMP	A	C	tmp.c->tmp.b->tmp.a-> tmp.c->tmp.b->tep.b->tep.a-> tmp.c->tmp.b->tep.b->tso.b->tso.a-> tmp.c->tmp.b->tep.b->tso.b->apv.b->apv.a-> tmp.c->tmp.b->tep.b->tso.b->apv.b->apa.b->apa.a-> tmp.c->tmp.b->tep.b->tso.b->apv.b->apa.b->ecp.c->ecp.b->

Table 8-14 provide for each area the initial CO (Init. CO), the target CO (Targ. CO) and the ways to follow to reach this last. These ways are the program (implementation of the "Path Prerequisites" algorithm) output.

The algorithm gives all paths leading to the current state to the target state and all control objectives to be achieved to reach the target state which is very difficult for maturity model's users to realize, especially when the activity interdependencies matrix is complex.

Conclusion

In this study we propose an algorithm called "Path Prerequisites" to define the IS risk management improvement plan. Based on the FA architecture, which is the most suitable maturity architecture for IS risk management maturity assessment, this algorithm looks for all the paths to reach a target state of Control Objectives (CO) based on a CO current state and then devise the list of those which must be achieved. After presenting the FA maturity model and its specificities, we give the description and "pseudo code" of the proposed algorithm. We then applied it to the information system risk management field. The proposed approach is very useful for determining the improvement plan for activities characterized by the interdependency between their elements.

The definition of an improvement plan is made for a single information system. This assumes that the improvement strategy is rather bottom-up. The improvement is done for each information system of the company then occurs the consolidation to have a global maturity. One possible perspective of this work is to focus on the top-down improvement strategy: defining the plan for improving information systems risk management maturity from a global maturity target.

Acknowledgement

The authors would like to thank their respective team members for offering helpful conditions to work.

Author's Contributions

Soumaya Amraoui: Writed the core of the manuscript.

Dr. Mina El Maallam: Coordinated the work and helped with writing issues.

Dr. Hicham Bensaid: Contributed in algorithm design, in programming stuff and helped with latex issues.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Alter, S.L. and S.A. Sherer, 2004. A general, but readily adaptable model of information system risk. Commun. Assoc. Inform. Syst., 14: 1-6.

Caralli, R. and M. Knight, 2012. Maturity models 101: A primer for applying maturity models to smart grid security, resilience and interoperability. Carnegie Mellon University, Software Engineering Institute.

Carlsson, S.A., S. Henningsson, S. Hrastinski and C. Keller, 2011. Socio-technical is design science research: developing design theory for is integration management. Inform. Syst. e-Bus. Manage., 9: 109-131.

Carvalho, J.V., A.L. Rocha, R. van de Wetering and A. Abreu, 2019. A maturity model for hospital information systems. J. Bus. Res., 94: 388-399.

Elmaallam, M. and A. Kriouile, 2013. Toward a Maturity Model Development Process for Information Systems (MMDEPSI). Int. J. Comput. Sci. Issues, 10: 118-125.

Elmaallam, M. and A. Kriouile, 2014. A generic process for the development and the implementation of is maturity models. Int. J. Comput. Sci. Issues, 11: 34-42.

Elmaallam, M. and A. Kriouile, 2015. Development of the ISR3M model for is risk management evaluation using the focus area structure according to the MMDPIS generic process. Transact. Machine Learn. Artificial Intel., 2: 106-106.

Elmaallam, M., H. Bensaid and A. Kriouile, 2019. A maturity model for assessing IS risk management activity considering the dependencies between its elements. Comput. Inform. Sci., 12: 98-111.

Herbsleb, J.D. and D.R. Goldenson, 1996. A systematic survey of CMM experience and results. Proceedings of the IEEE 18th International Conference on Software Engineering, Mar. 25-29, IEEE Xplore Press, Berlin, Germany, pp: 323-330. DOI: 10.1109/ICSE.1996.493427

Hevner, A.R., S.T. March, J. Park and S. Ram, 2004. Design science in information systems research. MIS Quarterly, 28: 75-105.

ISO, 2009a. ISO 31000:2009 risk management. Principles Guidelines Implementation.

ISO, 2009b. ISO guide 73:2009-risk management-vocabulary.

Iversen, J., P.A. Nielsen and J. Norbjerg, 1999. Situated assessment of problems in software development. ACM SIGMIS Database Adv. Inform. Syst., 30: 66-81. DOI: 10.1145/383371.383376

Koomen, T. and R. Baarda, 2006. TMap test topics. UTN Publishers.

Mayer, J. and L.L. Fagundes, 2009. A model to assess the maturity level of the risk management process in information security. Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management-Workshops, (NMW' 09), pp: 61-70.

Mettler, T., 2010. Thinking in terms of design decisions when developing maturity models. Int. J. Strategic Dec. Sci., 1: 77-88.

Olzak, T., 2008. A practical approach to managing information system risk.

Peffers, K., T. Tuunanen, M. Rothenberger and S. Chatterjee, 2007. A design science research methodology for information systems research. J. Manage. Inform. Syst., 24: 45-77.

Proenca, D. and J. Borbinha, 2016. Maturity models for information systems-a state of the art. Procedia Comput. Sci., 100: 1042-1049. DOI: 10.1016/j.procs.2016.09.279

Röglinger, M., J. Pöppelbuß and J. Becker, 2012. Maturity models in business process management. Bus. Process Manage. J., 18: 328-346.

Radut, C., 2009. The enterprise information system and risk management. Ann. Faculty Econ., 4: 1030-1034.

Rocha, A., 2011. Evolution of information systems and technologies maturity in healthcare. Int. J. Health Care Inform. Syst. Inform., 6: 28-36.

Rosemann, M. and T. de Bruin, 2005. Towards a business process management maturity model. Proceedings of the 13th European Conference on Information Systems, May 26-28, QUT, Germany, Regensburg.

Salvati, D., 2008. Management of information system risks.

Sienou, A., 2009. Proposition d'un cadre m´ethodologique pour le management int´egr´e des risques et des processus d'entreprise - a proposal of a methodological framework for the integrated management of risks and business processes. PhD Thesis.

Steenbergen, M.V., R. Bos, S. Brinkkemper, I.V. Weerd and W. Bekkers, 2010. The Design of Focus Area Maturity Models. In: Global Perspectives on Design Science Research, Winter, R., J.L. Zhao, (Eds.), Springer, Berlin, Heidelberg.

Stoneburner, G., A. Goguen and A. Feringa, 2002. Risk management guide for information technology systems. Nist Special Public., 800: 800-830.

Valentin, D. and F. Vasile, 2008. Erp implantation: Key factors of success and impact on performance. Ann. Faculty Econ., 4: 1353-1357.

Van Steenbergen, M., M. Van den Berg and S. Brinkkemper, 2007. A balanced approach to developing the enterprise architecture practice. Enterprise Inform. Syst., 12: 240-253.

Wade, M. and J. Hulland, 2004. Review: The resource based view and information systems research: Review, extension and suggestions for future research. MIS Q., 28: 107-142.

Weerd, I.V. and S. Brinkkemper, 2008. Meta-modeling for situational analysis and design methods.

Woodall, P., M. Oberhofer and A. Borek, 2014. A classification of data quality assessment and improvement methods. Int. J. Inform. Quality, 3: 298-298.

Algorithm 1

"Path Prerequisites" Algorithm

Purpose: The "Path Prerequisites" algorithm looks for all the paths to reach a COs target state based on a current Cos state, and then deduce the list of control objectives to achieve. The dependency constraints between the control objectives define a partial relation of order between them. They can be represented by directed graphs without loops, such as vertices and edges represent respectively the control objectives and the relation between them. One edge directed by an objective of control COi to another objective of control COj means that COi depends directly on COj. We consider the problem to provide from a current state of the control objectives, all prerequisites paths to reach a target state, while respecting the dependency constraints:

Input: *CurrentStatus*: the set containing the control objectives describing the current state,
TargetState: the set containing the CO target state.
Output: *Paths:* All paths leading to the current state to the target state,
List_CO_ to_ Achieve: all control objectives to be achieved to reach the target state.
Local variables: *P*: stack of paths to be explored,
Path: current path.

Algorithm:

1 **foreach** control objective COi **Input:**
2 *TargetState* **do**
3 *Path* =<COi >
4 *push*(*P*; (*COi*; *path*))
5 **while** *p is not empty* **do**
6 (*COj*, *path*) = *pop*(*p*)
7 **if** *COj is achieved* **then**
8 **if** *path* ∉ Paths **then**
9 *add*(*path, Paths*)
10 *add*(*COj, List_CO_to_Achieve*)

```
11      end
12 else
13      foreach objective COk (immediate previous of COj) do
14        /* If we reach a root
          control objective */
15        if COk has no predecessors then
16            if path ∉ Paths then
17                add(path, Paths)
18                add(COk, List_CO_to Achieve)
19            else Otherwise we rises along the graph
20                if COk is achived then
21                    path = add(COkpath)
22                    if path ∉ Paths then
23                        add(path, Paths)
24                        add(COk, List_CO_to_Achieve)
25                    end
26                    push(p, (COk, copy(path)))
27                end
28            end
29        end
30      end
31    end
32  end
33 end
```