

Research Article

Practical Experiences in the Use of Pattern-Recognition Strategies to Transform Software Project Plans into Software Business Processes of Information Technology Companies

C. Arevalo ^{1,2}, I. Ramos,^{1,2} J. Gutiérrez,^{1,2} and M. Cruz ¹

¹Department of Computer Languages and Systems, University of Seville, Seville, Spain

²Web Engineering and Early Testing Research Group, University of Seville, Seville, Spain

Correspondence should be addressed to C. Arevalo; carlosarevalo@us.es

Received 2 January 2019; Revised 18 February 2019; Accepted 26 February 2019; Published 2 May 2019

Guest Editor: Juan Carlos Preciado

Copyright © 2019 C. Arevalo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Business process management (BPM) is a strategic advantage for all kinds of organizations, including information technology companies (ITCs), which cannot stay out of the BPM approach. ITCs manage business processes like projects to create and maintain software. Although Project Management Systems (PMSs), such as Microsoft™ Project Server® (MPS®), are considered as non-process-aware information systems (Non-PAISs), they may be a source to generate processes. In this paper, we propose a reverse engineering approach, which uses patterns to transform software projects stored in MPS® legacy databases into software business processes. For this, we base on the model-driven engineering paradigm and deal with the time perspective of the processes. This kind of experiences are scarce or almost nonexistent, so we show the AQUA-WS project case study, which runs with MPS® as source system and software process modeling languages as target systems. ITCs can benefit from this research by gathering knowledge about perspectives of their processes that would otherwise be wasted, such as executed projects or expired documents used in Non-PAISs. This fact can become a key factor for ITCs, which can increase their competitiveness and reduce software costs, as part of the BPM lifecycle of continuous improvement.

1. Introduction

Competitiveness in the global economy is one of the most important challenges that companies and organizations must face up. Therefore, business process management (BPM) is a strategic advantage that all companies should consider [1, 2]. Information technology companies (ITCs) focused on software process management are intensely involved in these challenges [3, 4], although their business processes are more complex, variable, and unpredictable [5, 6] than those that take place in other industrial sectors. However, no ITC can afford to reject the BPM approach. In turn, model-driven engineering (MDE [7]) has been promoted during the last decades as a paradigm to solve the complexity associated with processes of software management. Object Management Group (OMG) Model-Driven Architecture (MDA [8]) is the major exponent of MDE in the field of software engineering. Henceforth, we will refer to

MDA in this article. ITCs have been working with this paradigm with certain level of success [9–12]. They often manage their operations using IT systems (ITs) that may be classified, among others, in two categories: (i) process-aware information systems (PAISs) [2], where the concept of process is well defined, comprising models and traces of instance executions that are stored in an *event log*; and (ii) non-process-aware information systems (Non-PAISs), which are often Legacy Information Systems (LISs) [13–16], containing a Legacy Database (LDB) that stores states of ITC transactions.

The BPM lifecycle of continuous improvement [2] proposes analyzing process executions against current process models. Business process discovery (BPD), concerning process mining [17–19] in the scope of PAISs, executes algorithms to construct new models from traces of process instances that are stored in the *event log*. New and old models may be compared, thus enabling business experts

to optimize processes. At this point, we wonder what happens with BPD in the scope of Non-PAISs that lack the *event log*. According to van der Aalst [20], LDBs store a lot of hidden evidence or knowledge related to process execution, so that they may be good sources to extract *Process* dimensions, even in the case of Non-PAISs. Regarding BPD from Non-PAISs, some authors, such as Adam et al. [21] and Zou et al. [22], propose techniques to recover processes, whereas other researchers, such as Pérez-Castillo et al. [10, 23, 24], Arevalo [25], and Arevalo et al. [26–28], use an MDA-based approach called Process Archeology (PA) for that purpose.

Business processes include different dimensions or perspectives. Among them, *Control Flow* is the essential one, although there are other process perspectives, e.g., *Time*, *Organizational*, *Resources*, *Data*, and *Cases*. In this paper, we show an MDA-based approach to obtain software business processes of ITCs. For this, we initially point to the hidden *Time Dimension* that may be scattered in some databases (see work by van der Aalst [20]) that ITCs use to manage their software lifecycle. To demonstrate the suitability of our approach, we have developed a case study with a public company named EMASESA (<http://www.emasesa.com>). This company is responsible for the cycle of water supply and sanitation networks in the city of Seville. It has developed a big modernization software project called AQUA-WS [29], which manages the transformation of old client-server LISs into a new Web-based integrated system. With regard to this case study, we highlight the fact that this project has involved multiple companies and organizations to carry out the following challenge: “*To develop an integrated and modular software solution to manage EMASESA.*” Basically, we have looked at the software process lifecycle used by different actors to develop the IT system of EMASESA. We do not focus on EMASESA business processes that run the cycle of water and sanitation networks, but to processes of software to solve its IT system. In this software project, we have played the role of main actors in the methodological field. The software lifecycle has been managed with NDTQ-Framework [30], following the Navigation Development Technique (NDT) by Escalona et al. [31]. In this case, NDT *Activities* are organized under a waterfall software lifecycle and Microsoft™ Project Server® (MPS®) has been the selected project management environment. We have applied our MDA-based approach to this case study in order to generate process models from source project plans that are stored in an MPS® database. This database suitably represents the *Time Dimension* of the project. For this reason, we have mainly faced up this dimension, although we are also interested in the others, since they may help to enrich process models. Besides, we have developed a specific Metamodel to extract *Time Dimension* from one or more source databases as well as to solve redundancy problems regarding *Activities* that are replicated in different LISs. We later discuss criteria for (i) selecting and classifying project tasks and (ii) mapping artifacts from projects onto process artifacts. Next, we analyze results, strengths, and weaknesses of processes. It must be mentioned that our approach can be applied in other software development projects involving

ITCs that use MPS® to plan and control processes since it would be enough to use other project plans. Moreover, any other organization whose business processes are project-oriented and uses MPS® could be a candidate to benefit from the proposal, even if it is out of the software sector. They could obtain representations of instances of their processes and process models, if the instances have well-classified activities within categories or types of activities. The use of other databases would also be possible but at the expense of generating new metamodels of them that capture the essence of the execution of processes in the organization.

In summary, the main objective of this paper is to obtain software business processes related to ITCs project plans since BPM is a strategic approach whose scope is significantly broader than simple software project planning and control. This approach will assist business experts in the implementation of the BPM approach and will facilitate the enrichment of processes with the data that are included in software project plans. We aim to propose new methods of process discovery from databases of Non-PAISs since Process Mining [17–19] performs the same from the PAISs *event log*. Our approach takes into consideration other existing methods in the literature to obtain processes from Non-PAISs. However, they focus on evidences related to different perspectives of processes, initially based on their *Time Dimension* as the basis for the heuristics of process generation. We have contributed with a method and a set of tools to facilitate the work of the experts in the software business. They will be able to reuse the hidden knowledge about software business processes stored in databases, which would otherwise be forgotten or wasted. Additionally, it should be added that we expect better levels of efficiency and effectiveness when compared to the manual analysis of those processes.

The rest of the paper is organized as follows: Section 2 presents related work. Section 3 shows the main topics of our approach to model the *Time Dimension* of software projects. Section 4 shows an MDA-based proposal for taking out processes from databases of Non-PAISs taking into account different roadmaps; a specific MDA-based business process discovery roadmap is developed, which allows transforming project plans from MPS®. Section 5 analyzes how this roadmap behaves in a real case study, the AQUA-WS project, where results, advantages, and limitations of the approach are discussed. To finish, Section 6 states conclusions and future lines of research.

2. Related Works

Nowadays, ITCs are still working with LISs. Some LISs are PAISs, whereas most of them are Non-PAISs. Process Mining Techniques [17–19], are suitable to carry out business process discovery with *event logs* existing in PAISs, but they are not a choice to extract processes from Non-PAISs due to the lack of *event log* files. OMG Architecture-Driven Modernization (ADM) [32], which is an OMG’s MDA-based [8] proposal for the modernization [13–16, 33] of LISs, includes reverse [34, 35] and forward engineering roadmaps from an old source system to a new target one.

ADM uses Abstract Syntax Tree Metamodels (ASTM) and Knowledge Discovery Metamodels (KDM) to extract knowledge from source systems. We are more interested in reverse [34, 35] engineering roadmaps that may help us to discover processes from source LIS artifacts, which may be (i) source code; (ii) graphical user interfaces (GUI); or (iii) databases. We have focused on the last one because databases are the most stable artifacts in a LIS. Since MDA and ADM are general standards that do not focus on process discovery field, we have needed to explore other literature regarding database reverse engineering as well as specific proposals with the aim to discover processes from Non-PAISs.

There is a lot of research work related to databases reverse engineering, but we have just selected some of them due to the likeness that they keep to our work. We would like to mention the work by Cleve et al. [36], who propose data reverse engineering using System Dependency Graphs (SDG) that analyze Data Manipulation Language (DML) sentences of Structured Query Language (SQL) that are scattered in the application code. They propose a new database schema by adding new candidate and foreign keys that may be inferred from SDGs related to the DML SQL embedded in the application code. Additionally, it is also remarkable Arevalo [25] and Arevalo et al. [26, 27] dealing with reverse engineering databases (i.e., relational tables, declarative constraints, and triggers) to define business Event-Condition-Action (ECA) rules over processes, expressed by means of Unified Modeling Language (UML [37]) and Object Constraint Language (OCL). Similarly, but not oriented to processes, it is relevant the work by Cosentino and Martínez [38], who also extract UML classes and OCL rules from tables and triggers. Finally, the proposal of Zanoni et al. [39] point to the evolution of software systems by pattern detection for conceptual schema recovery in data-intensive systems.

As we are interested in BPD field, we have selected some research works by Pérez-Castillo et al.: (i) those which propose Modernization Approach for Recovering Business Processes from Legacy Systems (MARBLE) [10, 23, 24] as a framework that extends the ADM standard and (ii) Pérez-Castillo et al. [33], who propose recovering Web Services from databases. These studies [10, 23, 24, 33] work as ADM, with KDM and ASTM, recommending different steps with KDM to discover business processes. The authors point to relational database DML sentences to propose new relational database schemas using ideas such as those included in the aforementioned work by Cleve et al. [36]. The BPD approach can generate business processes of different sizes and structures that may be characterized by connectivity, density, and separability of artifacts. Generated processes may recurrently present disadvantages regarding quality parameters such as comprehensibility and modifiability [40]. Process refactoring [40–42] includes techniques to write alternative process instances by adding, deleting, or redistributing existing process artifacts. Artifacts may be activities, gateways, events, or control flows. The refactored process is a new process instance with the same semantic as the source process instance, which is generated by applying

some rewriting rules. Refactoring processes quality is evaluated using artifact-based measurements. Caivano et al. [40] evaluate the process quality perceived by experts (human-perceived measures). They compare both, artifact-based and human-perceived quality measures to conclude that “Process refactoring is worthwhile so that humans reach better levels of comprehensibility and modifiability.”

In order to manage the software lifecycle, business experts use General Process Modeling Languages (GPMLs) [43–45], such as Petri Nets, Business Process Execution Language (BPEL), Business Process Model and Notation (BPMN) [46], and Event-Driven Process Chain (EPC) or Unified Modeling Language Activity Diagrams (UML AD) [37], together with other specific Software Process Modeling Languages (SPMLs), such as Software and Systems Process Engineering Metamodel (SPEM [47–49]), Software Engineering Metamodel for Development Methodologies [50], and Essence-Kernel And Language For Software Engineering Methods (Essence [51, 52]). Besides, we have taken into account NDTQ-Framework [30], because it is used in our case study: AQUA-WS Project [29]. Bonnet et al. [53] consider BPMN as the process modeling leading standard between users and business experts, which is increasingly being utilized in the software field [43–45].

Although we are interested in different process dimensions, this paper particularly addresses the *Time Dimension* to generate software business processes for ITCs. Processes may be used by software experts with business process management systems (BPMSs) in a BPM lifecycle of continuous improvement. There are many approaches defining aspects regarding the *Time Dimension* of processes, but for the purpose of this study, we have selected those that model *Time Rules* on projects. Among other works, Flores and Sepúlveda [54] analyze *Time Rules* in project plans with the aim to depict them as BPMN processes. They propose time patterns, even though this solution overloads the main *Control Flow* of the process by adding a lot of imperative artifacts. Furthermore, Time-BPMN [55] is a clean and elegant proposal that extends the language with *Time Rules* by means of incorporating new decorators that are not still supported by BPMN 2.0 standard [46]. Cheikhrouhou et al. [56] extend the original Time-BPMN proposal by introducing new *Time Rules*.

According to the work by van der Aalst [20], databases hide knowledge related to processes, so that they may be good sources to extract process dimensions from databases of Non-PAISs. This paper constitutes a theoretical foundation that allows using databases as a source to construct *event logs* in systems that lack them, so that generated logs can feed Process Mining Techniques [17, 18]. Building on this framework [20], the work by González López De Murillas et al. [57] is an initiative that uses database Redo logs as a source artifact to generate an *event log*. The second initiative by González López De Murillas et al. [58] goes on in the same direction, i.e., they propose a metamodel and tools to connect databases with Process Mining [17–19].

The aforementioned works, regarding BPD from Non-PAISs, only generate some aspects of them. Nonetheless, as

results may appear to be poor in the eyes of business experts, they are not widespread. We expect that our proposal will be able to generate richer results, looking at the specific field of software lifecycle management and capturing dimensions (initially *Time Dimension*, but extendible to others such as *Organizational*, *Resources*, *Data*, and *Cases*) of processes that these Non-PAISs may hide. Taking into account the previous related work, we suggest a reverse engineering approach, composed of an MDA [8] infrastructure and heuristic methods, which allows turning projects stored in legacy databases into software business processes of an ITC. In comparison with some approaches cited above, which just use different transformation steps between ASTM and KDM, our heuristics do not use KDM and initially centers on *Time Dimension* of processes that may be scattered in databases of ITCs, which otherwise would be wasted for BPM purposes. We compare (cf. Table 1) our approach, initially tested with the AQUA-WS [29] project, with above initiatives that are close to process discovering.

Other authors' approaches, which use SDG [36], ASTM [10, 23, 24], or KDM [10, 23, 24], gather all kinds of Non-PAISs database artifacts that do not focus on process dimensions. The approaches that use SDG [36], ASTM [10, 23, 24], or KDM [10, 23, 24] collect all types of artifacts existing in databases of Non-PAISs, although they do not face the dimensions of the processes as a heuristic basis for generating them. There are few or almost nonexistent experimental cases that utilize these proposals to extract processes. Additionally, the results obtained do not go beyond deriving conceptual database schemes or poor approximations to real processes. In this paper, we extend our initial proposal [25–27] and show a detailed framework that addresses the selection of database artifacts that are closely related to process execution traces. This framework supports different roadmaps depending on source systems and selected target languages that software experts use to describe their processes. To the best of our knowledge, we have not found out approaches in the literature as our proposal, i.e., focusing on metamodels (cf. † in Table 1) concerning process dimensions that are scattered and hidden in legacy databases.

3. Time Dimension of Business Processes

ITCs as many other organizations are introducing the BPM approach for improving [43–45, 53] their software business processes, which means a key factor to become more competitive. As previously mentioned, processes have different perspectives or dimensions. The main one is represented by *Control Flow* of *Activities*, although the *Information* perspective may also be depicted as *Data Flows*. Besides, *Cases*, *Organizational*, *Resource*, and *Time Dimension* may also be represented in relation to business processes. In previous work [28], we principally focused on analyzing the *Time Dimension* of processes. As *Time Perspective* is concerned, a *Time Rule* is a subtype of *Business Rule* [59], which is well defined in turn by several authors, such as Ross [60], Wagner [61], and Cheikhrouhou et al. [56]. They suggest different classifications.

Orchestrations [46] are private business processes executed in an organization, but there are more complex ones, such as Choreographies [46] or Inter-Organizational Business Processes [56, 62]. In this paper, we have addressed software business processes executed by ITCs as Orchestrations. We have identified *Time Rules* classes that usually constrain a process and we have also developed an approach [28] that proposes a *Time Rule Taxonomy* [28] concerning Orchestrations, which is defined in a Process Metamodel (cf. Figure 1(a)). This Metamodel has a minimum set of classes to reach a good level of interoperability [43, 63] between GPMLs and SPMLs. Time rules [28] are defined as OCL constraints. Figure 1(b) depicts an example of *Time Rule*: “The Start to Finish” *Time Dependency (TD)* between a *Successor Activity* and a *Predecessor Activity*.

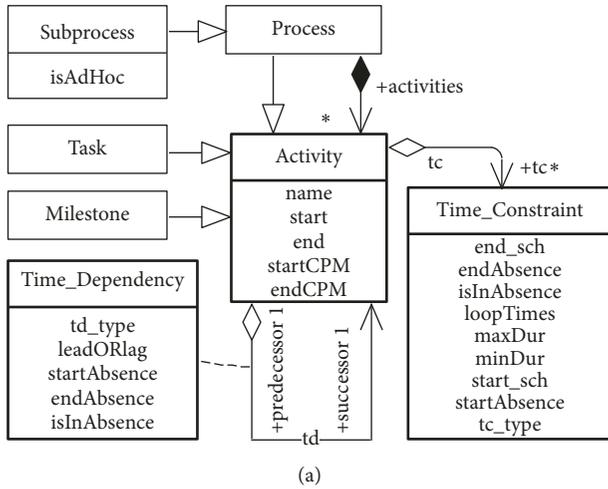
Classes of Metamodel [28] support the definition of *Processes*, which are composed of a set of *Activities*. *Sub-process* is a subtype of *Process*, and the *Activity* class is specialized with *Task* and *Milestone* subclasses. Furthermore, *TC* (*Time_Constraint* class) and *TD* (*Time_Dependency* class) may also be defined. Regarding attributes (i) *Sub-process* maybe Ad Hoc (*isAdHoc*), that means, its *Activities* run in parallel, without additional restrictions; (ii) *Activity* has *name*, *scheduled* (*startCPM* and *endCPM*) and *executed* (*start* and *end*) *events*; (iii) *Time_Constraint* class contains the attributes *TC type* (*tc_type*), maximum (*maxDur*), and minimum duration (*minDur*), constraints dates (*start_sch* and *end_sch*), attributes concerning *Absence Constraint* (*isInAbsence*, *startAbsence*, and *endAbsence*), and finally the number of loops (*looptimes*) that an *Activity* may run in terms of *Cardinality TCs*; and (iv) *Time_Dependency* class includes properties *TD type* (*td_type*), whether the *Activity* has an *Absence Dependency* with respect to *Predecessor Activity* (*isInAbsence* and, optionally, time interval [*startAbsence*, *endAbsence*]). *Events* of a *Successor Activity* may be constrained with a time lapse (*leadOrlag*) in terms of *Predecessor Activity* event, sometimes a *lead* (*leadOrlag* is negative) and in other cases a *lag* (*leadOrlag* is positive).

The taxonomy [28] includes the following elements: (i) *Time Constraints (TCs)*, which only affect an *Activity* in a *Process* and (ii) *TDs*, which involve rules between two *Activities*. Both of them regulate the *start* and *end events* of *Activities*. Since we initially pay attention to software projects carried out by ITCs, we have selected, from the referenced *Taxonomy* [28], *Time Rules* that may be found in most Project Management Systems (PMSs) (cf. Tables 2 and 3). We detail the rules they comprise as follows:

- (i) *Time Constraints (TCs)* are classified into (a) *Duration* of *Activities*; (b) *Fixed* or *inflexible start* and *end events*; (c) *Flexible start* and *end events*; (d) *Cardinality*, which establishes constraints over the loop iterations and duration; and (e) *Absence Constraint*, which avoids the execution of an *Activity*.
- (ii) *Time Dependencies (TDs)* involve *Predecessor* and *Successor Activities*. They can be classified as follows: (a) *Rules defined in Allen's Interval Algebra* [64] and (b) *Absence TD*, which avoids the execution of a *Successor Activity* depending on *Predecessor events*.

TABLE 1: Modernization initiatives of Legacy Information Systems (LISs).

Initiative	Source artifacts		Approaches used to discover hidden knowledge from LISs						Results			
	LDB	App code	GUI	SDG	ASTM	KDM	MM [†] concerning process dimensions	UML	OCL	ECA rules	LDB schema generation	Business process discovery
ADM [32]	☑	☑	☑					☑			☑	
Cleve et al. [36]	☑	☑		☑							☑	
Cosentino and Martínez [38]	☑							☑	☑		☑	
Pérez-Castillo et al. [10, 23, 24]	☑	☑	☑		☑	☑		☑			☑	☑
Arevalo et al. [26, 27]	☑				☑			☑	☑	☑	☑	☑
Arevalo [25]	☑				☑		☑	☑	☑		☑	☑
Our current approach	☑				☑		☑	☑	☑		☑	☑



Context Time_Dependency
 inv: self.td_type = 'SF' implies

$$self.predecessor \longrightarrow select \left(\begin{array}{l} P | \\ \left(\begin{array}{l} self.successor.end \\ \leq (P.start + leadORlag) \end{array} \right) AND \\ \left(\begin{array}{l} self.successor.endCPM \\ \leq (P.startCPM + leadORlag) \end{array} \right) \end{array} \right) \longrightarrow notEmpty()$$

FIGURE 1: Process Metamodel [28] to capture Time Dimension. (a) Process Metamodel Class Diagram [28]. Project decomposition into Activity hierarchies and Time Rules: TCs and TDs. (b) An OCL sample TD included in Process Metamodel [28]; “Start to Finish “SF” Rule.”

TABLE 2: TCs supported by PMSs.

Category	Time Constraint
Duration	FIXD: Fixed Duration FLEXD: Flexible Duration
Inflexible events [†]	MSO: Must Start On, MFO: Must Finish On
Flexible events [†]	ASAP: As Soon As Possible $\longrightarrow S_{ASAP}$ and F_{ASAP} ALAP: As Last As Possible $\longrightarrow S_{ALAP}$ and F_{ALAP} NET: Not Earlier Than $\longrightarrow S_{NET}$ and F_{NET} NLT: Not Later Than $\longrightarrow S_{NLT}$ and F_{NLT}

[†]S (start) or F (finish) events.

TABLE 3: TDs supported by PMSs.

Category	Time Dependency
Allen’s algebra rules	SS: Start to Start SF: Start to Finish FS: Finish to Start FF: Finish to Finish

Arevalo et al. [28] include detailed definitions of each rule and OCL formulation.

4. An MDA-Based Approach to Generate Software Business Processes from Non-PAISs Legacy Databases Used by ITCs

ITCs are increasingly using the BPM approach to manage [43–45, 63] their software business processes, although they still use LISs that may be Non-PAISs as follows: (i) PMSs (such as MP®, MPS®, or Redmine®), which allow software experts to plan and control projects; (ii) Enterprise Content Management Systems (ECMs) (such as Alfresco® or Sharepoint®), which allow document management, collaboration, and subscriptions; (iii) a collection of ITs, e.g., Enterprise Resource Planning Systems (ERPs) (such as SAP® or Microsoft Axapta®), Customer Relationship Management Systems (CRMs) (such as Oracle Siebel®), and Supply Chain Management Systems (SCMs) (such as Kinaxis® or Blue Ridge®); and finally (iv) Tailor-Made Software

Systems. Furthermore, BPMs are specialized ITs that may be integrated with other classic Non-PAISs. BPMs support the BPM lifecycle of continuous improvement. Therefore, we have focused on how to reuse hidden knowledge of processes stored in databases, which would otherwise be forgotten and wasted.

This section shows our approach to generate software business processes from Non-PAISs legacy databases used by ITCs. Our proposal is an MDA-based framework that allows multiple reverse engineering roadmaps. Each roadmap implies a source system and a target system. We focus on (i) legacy databases as source systems and (ii) process modeling languages used by software experts as target systems. The first section establishes the architecture of our MDA-based solution. We show common aspects to all roadmaps. The second section focuses on ITCs as project-oriented organizations, who manage software project plans with MP[®]. We have developed a specific roadmap to transform software project plans stored in an MPS[®] legacy database into software business processes. Transformation heuristic is based on extensions of the Process Metamodel [28] (cf. Section 3). We will show detailed mapping rules with an algorithm and tables.

4.1. Architecture of Our MDA-Based Solution. We have pointed at specific processes of ITCs for software lifecycle management. Therefore, we have analyzed databases from diverse ITs (among others, MPS[®], RedMine, Alfresco, and Sharepoint) gathering structures and rules concerning different dimensions of this kind of processes, because such databases hide a lot of knowledge generated by each ITC [20]. On these databases, we have studied the ability to extract structures and rules that are related to the main process dimensions, such as *Time*, *Resources*, and *Cases*. After the analysis, we come to the conclusion that (i) PMSs lay a strong foundation for *Time Dimension*, although they include definitions for *Resource* management; (ii) ECMs are suitable ITs for *Resource* management and also entail some *Time Rules*; and (iii) ERPs, CRMs, SCMs, and Tailor-Made software may involve rules concerning all process dimensions. Market or Standard systems are better choices than Tailor-Made ones since we can generate processes for many organizations that use the same system by utilizing the same roadmap. Initially, we have focused on PMSs, so we have analyzed *Time Rules* they support. Table 2 shows *TCs*, and Table 3 represents *TDs* that are usually included in PMSs.

Our approach is based on MDA [8] concepts. Figure 2 depicts a generalized MDA-based architecture to generate software business processes of ITCs from some databases of Non-PAISs. There may be different roadmaps to generate processes depending on the selected source and target systems. Each roadmap represents a concrete path that allows transformations from some source database artifacts into target Business Process Modeling Languages (BPMLs) that may be used to manage software business processes (GPMLs or SPMLs). The prerequisites for a candidate source database are (i) the source Non-PAIS (such as some PMSs,

ECMs, ERPs, CRMs, or SCMs) must be used to manage software business processes and (ii) database must include some relevant artifacts (tables, constraints, and triggers) concerning the *Time Dimension* of software business processes of ITCs. A candidate database stores the hidden knowledge regarding processes of ITCs that we are looking for. Software experts choose their favorite SPMLs or GPMLs. With the aim of achieving greater interoperability, we propose to carry out the reverse engineering of processes up to our Metamodel [28], as an intermediate result that is platform independent, which means processes do not depend on the concrete syntax of any language. This Metamodel shares semantics (common classes and associations in process models) with the main SPML or GPML metamodels, which will allow us to easily export results through XML standard data exchange formats.

Our MDA [8] infrastructure consists of a set of metamodels at different levels of abstraction and transformations. Each model conforms to its metamodel, then metamodel rules are applied to each model. Transformations are based on heuristics in terms of our core Process Metamodel [28] (cf. Figure 1(a)). They offer interoperable models with concrete BPMLs. The approach could also be extended to capture other process dimensions from databases, such as *Resource*, *Organizational*, *Case*, or *Data*.

The main components of the MDA-based proposal are (i) source system, (ii) target system, and (iii) MDA [8] transformations.

- (i) *Source System.* We have mainly looked at databases; therefore, it is important to know the data models that conform to *correspondent* metamodels. A Platform Specific Metamodel (PSM [8]) allows formalizing models within each source system. We must find database artifacts that are closely related to *Time Perspective* of processes, by means of analyzing reduced views that show task models, involving *Activities*, *Milestones*, and *Time Rules* gathered as hidden knowledge from source databases. These task models are represented on the technological platform corresponding to their Database Management System (DBMS), which is commonly a Relational Database Management System (RDBMS). That is why we need both generic metamodels (GASTM) and specific (SASTM) metamodels.
- (ii) *Target System.* For software business processes of ITCs, the target system may be a BPML, either SPML (such as SPEM [47–49], ISO/IEC 24744 [50], Essence [51, 52], and NDTQ–Framework [30]) or GPML (such as BPMN [46] and UML AD [37]). These languages share some common characteristics along their process metamodels. They comprise the computer-independent (CIM [8]) level where ITCs business experts work.
- (iii) *Heuristics to Generate Hidden Knowledge of Business Processes.* We propose a Model-To-Model (M2M) procedure [25, 27] that uses the previous MDA [8] infrastructure to explore databases. We have based

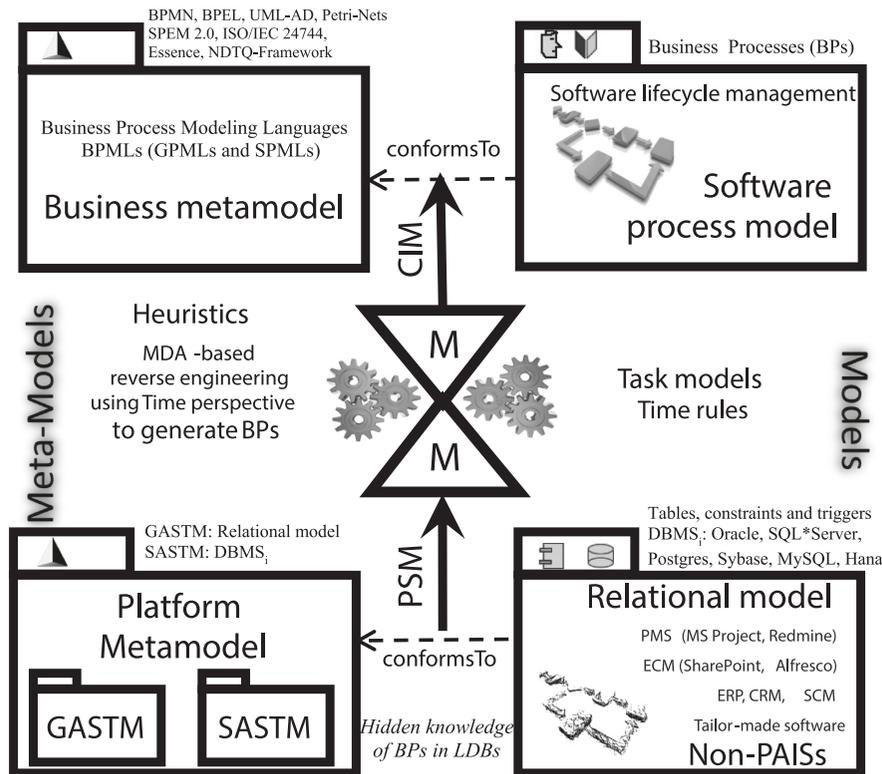


FIGURE 2: MDA-based infrastructure to generate software business processes (BPs) from LDBs used by ITCs. Candidate roadmaps to transform LDB task models into BPs.

the heuristics of process generation on identifying mapping among existing structures and rules in the PSM platform that correspond to classes and associations of our Process Metamodel to capture *Time Dimension* in a platform-independent metamodel (PIM [8]). This Metamodel has been proposed to extend GPML, such as BPMN, with time [28] (cf. Figure 1(a)) and also to support constraints of Tables 1 and 2, as well as some others that PMSs do not frequently use. The Metamodel has a minimum set of classes that share previous BPMLs (SPMLs and GPMLs); consequently, it is not difficult to match this point with the desired BPML.

At this point, we may also show a procedure (cf. Figure 3(a)) to generate software business processes from databases of ITCs using our MDA-based proposal. Figure 3(a) illustrates the general steps to follow:

- (i) The activity “To establish MDA generic infrastructure” is executed as a subprocess that explodes in detail tasks in Figure 3(b), including “To set a basic PIM Process Metamodel” and “To extend Metamodel with Time Rules,” which concern Process Metamodel [28] of Figure 1(a). If this proposal is used to generate processes from different database sources, we need a flexible and extensible *PIM Process Metamodel* that allows merging *Activities* and solving possible inconsistencies or

redundancies. In our first case study, it has not been necessary to prepare and execute this activity yet. This first procedure is common to all possible roadmaps from a database of an ITC to a BPML.

- (ii) Figure 3(c) depicts “To Stablish MDA specific infrastructure depending on the roadmap” into *Activities* “To extract Task Process Metamodel from Source Legacy Database” and “To Extend Target Process Metamodel with PIM Metamodel Rules”. The former has to release database views including artifacts concerning business processes, their decomposition into *Activities*, and *Time Dimension* linked to such *Activities*. Prerequisites to use database of an ITC are associated with study database views (tables, attributes, constraints, and triggers; see Figure 4, which represents the database view in our case study) concerning hidden evidence of process executions; that is, it looks for instances of artifacts conforming to classes of *PIM Process Metamodel* [28] (cf. Figure 1(a): *Processes, Activities, Time, and Resource Rules*). The latter extends selected BPML with *Time Dimension* of *PIM Process Metamodel* [28] (Figure 1(a)). Both *Activities* are necessary for each specific roadmap. Efforts may be best monetized if selected source Non-PAISs and target BPMLs are widely used by ITCs to manage their software business processes. That is to say, a good roadmap goes from LISs with high market share to standard BPMLs that are well accepted by

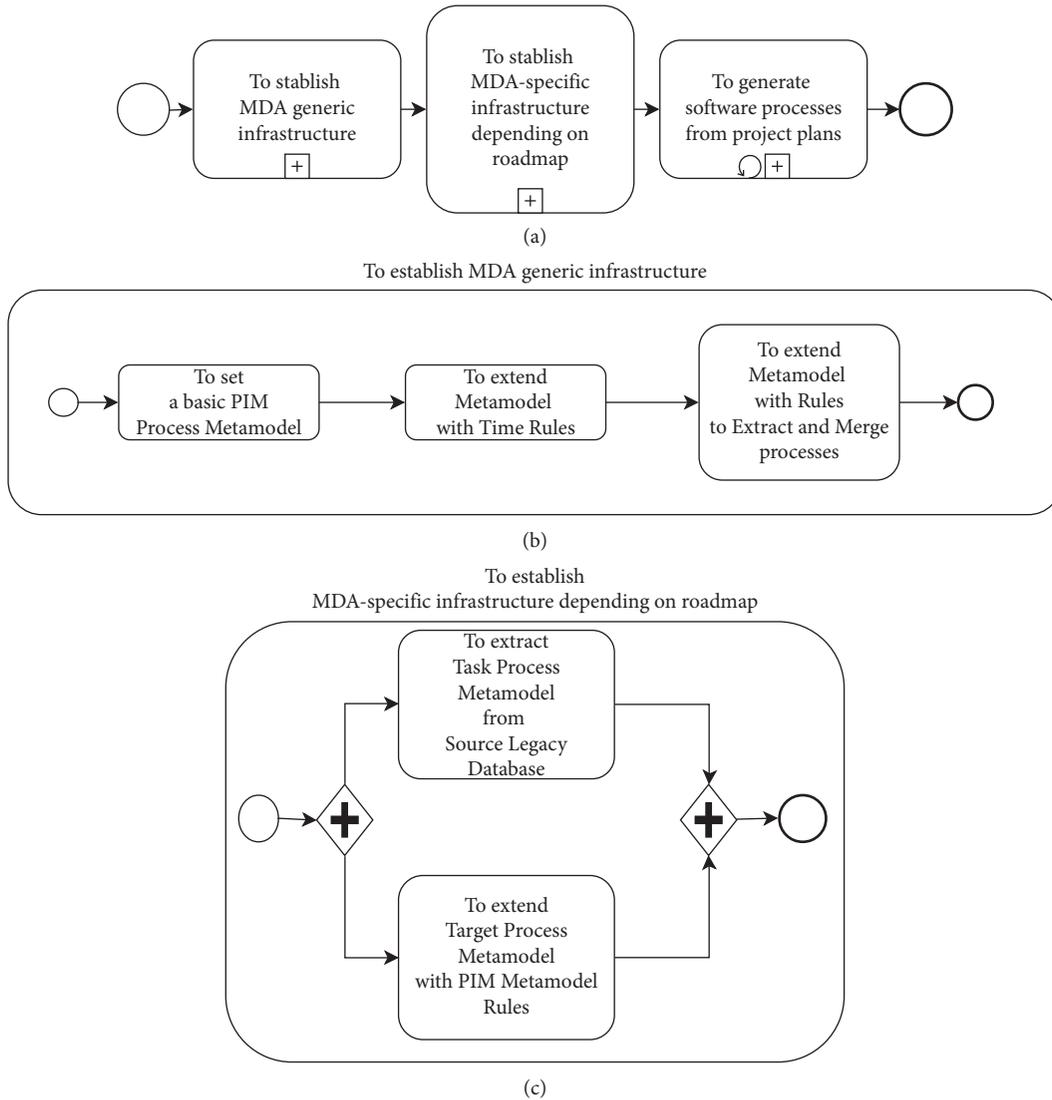


FIGURE 3: MDA-based procedure to transform projects stored in databases of PMSs into Software Business Processes of ITCs. It takes into account the common activities of all roadmaps and specific activities of a concrete roadmap. (a) Procedure to execute Software Business Process generation roadmaps. (b) Common activities of the MDA-based approach. Preparation of Process Metamodel [28] with Time Rules, as a Platform-Independent Metamodel (PIM) to execute all roadmaps. (c) Activities to execute each candidate roadmap, conditioned by source and target platforms.

business software experts so as to manage the software lifecycle.

- (iii) Using this MDA infrastructure (common and specific to each roadmap), “*To Generate software business processes from project plans*” (Figure 3(a)) may be executed once or more times in a loop, in order to obtain software business processes from available sources.

4.2. An MDA-Based Roadmap to Transform Software Project Plans Managed with Microsoft Project into Software Business Processes. Regarding PMSs, Microsoft™ Project® (MP®) and Microsoft™ Project Server® (MPS®) are market products used throughout the world in organizations whose business processes are project-oriented; ITCs are not an

exception, and MPS® has been used for many years by most of IT business experts all over the world, so we expect that our effort may be rewarded with the capability to generate a larger number of instances and models of processes available to these experts. Furthermore, we have analyzed RedMine and Alfresco, which are commonly used by ITCs, among other many other organizations out of the software field, so these source systems could also be in new roadmaps to generate processes of software. All of them are Non-PAISs whose project repository is stored in a legacy database. We have analyzed database metamodels of these products, but we focus on MPS® legacy databases, which are supported by four kinds of Microsoft™ SQL*Server® instances: *Drafts*, *Published*, *Archive*, and *Reporting*. We have chosen *Published* instance, as it has the same structure as *Draft* instance, but stores the detailed information on tasks, links, and all

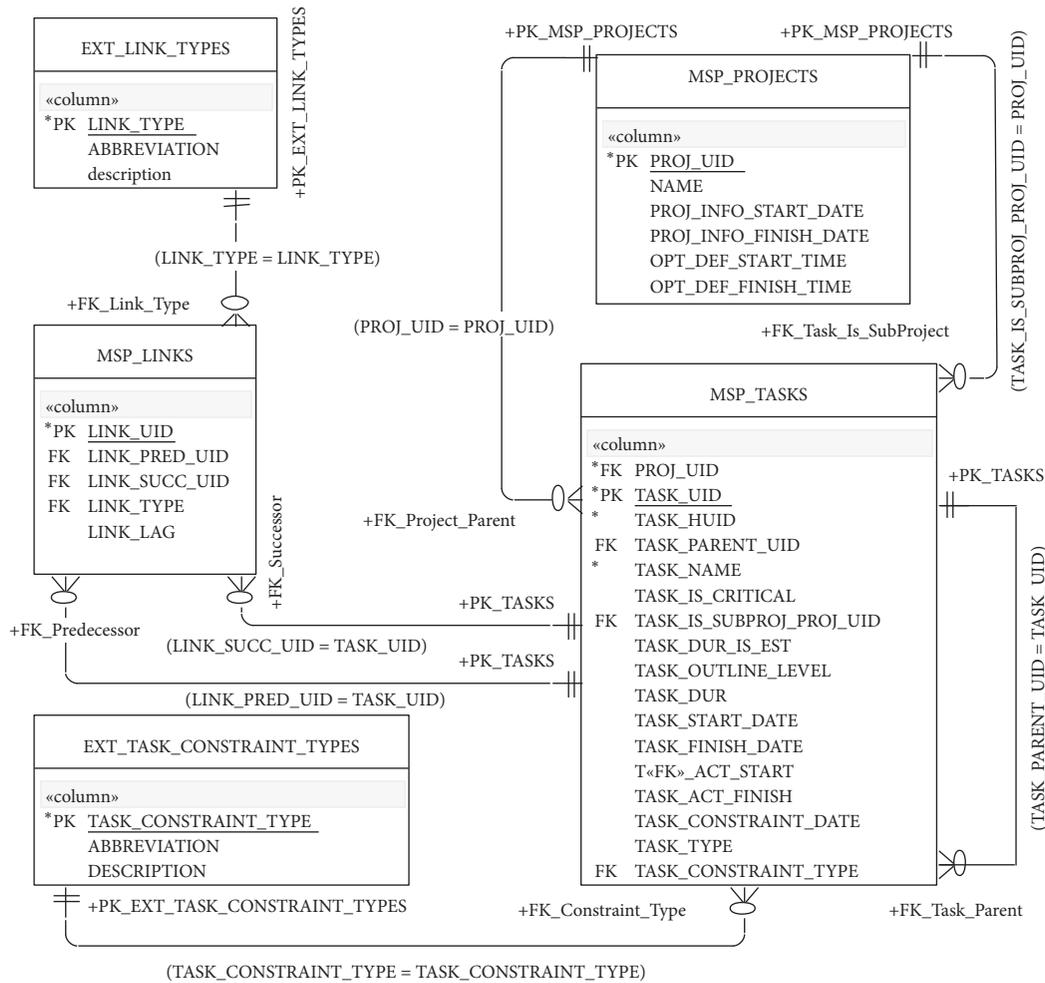


FIGURE 4: MPS® database task model (Published instance). Database view concerning *Time Dimension* of projects plans.

kind of constraints that the system provides for planning and replanning projects. Published and Archive instances are derived from Drafts and Published ones.

We have developed an initial roadmap shown in Figure 5 that is a specialization of Figure 2 (suitable for all roadmaps). MPS® legacy database is the selected source system. As experts choose their favorites software process languages, we do not want to constraint to only one language to describe software business processes, so the target system that will be our platform-independent Process Metamodel [28]. This way, it will be possible to use different BPMLs as targets, because artifacts (classes and rules) of our Metamodel exist in most of BPML metamodels commonly used by software experts. Generated process models can be represented with SPMLs or GPMLs that are widely used by ITCs. Powerful target systems are BPMN and the Workflow Management Coalition (WfMC) XML Process Definition Language (XPDL). They will allow the use of the World Wide Web Consortium (W3C) Extensible Markup Language standard (XML) to exchange serialized processes (schemas and instances). BPMN allow serializations with .xsd and .xmi XML formats and XPDL with .xpdl format. Most of BPMLs (GPML and SPML) supporting tools allow the use of .xsd,

.xmi and .xpdl exchange formats to serialize processes. In summary, we have aimed to generate instances and process models from project plans focusing on task structures and Time Rules. As future work, we will be able to use further source systems, such as other PMSs, ECMs, ERPs, CRMs, SCMs, or Tailor-Made software. Besides, we will propose roadmaps to specific SPMLs or GPMLs targets, used by ITCs that work with this type of systems. In this case study, we have considered the following aspects of source systems, target systems, and heuristics to generate business processes:

- (i) *Source System MPS®*. Database prerequisites to be a source for generating software business processes (cf. Section 4) involve exploring a metamodel regarding projects structure and their *Time Dimension*. Database Task model in Figure 4 allows us to explore a lot of projects in many ITCs that manage the software lifecycle. The source metamodel must support SQL*Server, which conforms to ISO SQL 1992, so a generic relational metamodel (GASTM) for this standard and a specific metamodel (SASTM) for *Microsoft™ SQL*Server®* [26] are needed. Task model (Figure 4) is taken out from

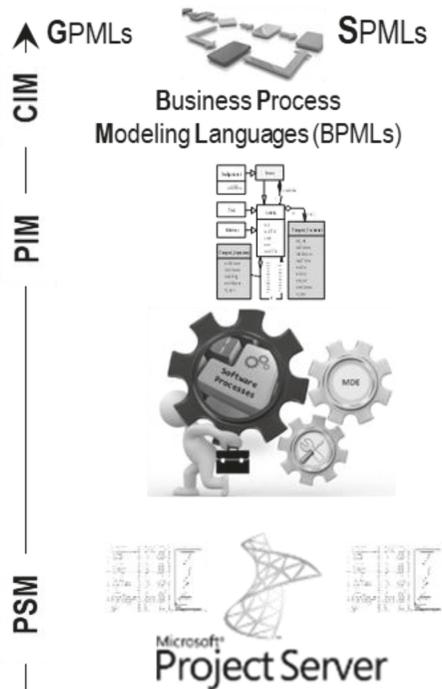


FIGURE 5: MDA-based roadmap from MPS® database to PIM Process MM [28] (as a roadmap case of Figure 2).

the MPS® Published instance. Relational *tables* and *foreign keys* are displayed below:

- (a) *MSP_PROJECTS Table*. It stores information concerning projects.
 - (b) *MSP_TASKS Table*. It displays rows that show subordinated tasks of a project through the *FK_Project_Parent* foreign key. Moreover, if a task is a subproject, then it may be scheduled as an external project (*FK_Task_Is_Subproject*). Consequently, *Activities* may be organized hierarchically with task groups as parent tasks and child tasks. These relationships are expressed through *FK_Task_Parent*. The task table enables defining due dates over task *events* (*start* and *end*) as well as setting task duration (fixed or estimated). *TCs* of Table 2 are supported. *EXT_TASK_CONSTRAINT TYPES* table constitutes the enumeration of *TCs*.
 - (c) *MSP_LINKS Table*. It let us identify relationships between *Predecessor* and *Successor Tasks*. *TDs* of Table 3 are supported. *EXT_LINK_TYPES* table represents the enumeration of *TDs*.
- (ii) *Target System*. We have used our metamodel [28] (cf. Figure 1(a)), which holds extended *Time Rule*'s semantics that has not been supported by languages such as BPMN yet. This target system works as a platform independent of technology. Gantt or PERT diagrams just allow planning or executing projects. BPMLs are increasingly used for modeling software business processes. BPMLs, such as SPMLs [43, 63] and GPMLs [43–45], are quite more powerful for

modeling other dimensions, such as *Control Flow*, *Organizational*, *Data*, and *Resources*, as well as for being the gateway not only to model but to execute and audit those processes in a cycle of continuous improvement. From the point of view of software lifecycle management, focusing on processes gives the expert a much broader scope than just pointing to planning projects.

- (iii) *MDA-Based Heuristics to Generate Business Processes*. The heuristics highlight the identification of *Time Rules* concerning projects (*TCs* and *TDs*). In this paper, we have solved mappings between the source system and target Time-based Process Metamodel [28], which are M2M transformations. Table 4 shows correspondences between the properties of the source system tables (project plans) and the attributes of the metamodel classes of processes with temporary rules. They aim to extract business processes regardless of the platform or target language:

- (a) Table 4(a) depicts mappings of a *Project* onto a *Business Process*. MPS® allows to manage project hierarchies; this means that a *parent project* may have an activity that also is a *subproject*. In this case, there are two instances of projects, the parent one and the child subproject so a link for association “*is_a_Activity*” exists.
- (b) Table 4(b) contains the details of mapping *project tasks* onto *Process Activities (+activities)*. *Single tasks (activity is_a Task or is_a Milestone)* are mapped as *Activity subtype*. Another activities hierarchies are “*Nested Tasks*” that means a *parent activity* is decomposed into *child activities*. If a project task were decomposed into more detailed tasks, then the corresponding *Activity* would be also a *Subprocess* and a *Subprocess* would be a subtype of a *Process*. These generalization relationships are solved in Table 4(c). *Subprocesses* are *AdHoc*, meaning that the execution flow runs in parallel for the group of tasks, without additional restrictions.
- (c) Table 4(d) includes mapping rules for those cases where a task is an external subproject. It is mapped as a subprocess.
- (d) Mappings of *Duration Rules* allowed by the source system are detailed in Table 4(e). *Fixed (FIXD)* and *Flexible Duration (FLEXD)* rules are expressed by OCL constraints.
- (e) Table 4(f) shows mapping criteria for *TCs* over *Activity events*, including *Flexible* and *Fixed* rules concerning the *start* and *end* of an *Activity* (*TCs* of Table 2 are supported).
- (f) Finally, Table 4(g) depicts mappings for *TDs* between a *Predecessor* and a *Successor Activity* (*TDs* of Table 3 are supported).
- (g) Table 4(e)–4(g) include the column “*OCL constraints*,” concerning *TCs* and *TDs*, thus generated processes inherit *Time Rules* that are

TABLE 4: Mappings from the source system onto the target system. Source: MPS® legacy database Metamodel (cf. Figure 4) → Target: Process Metamodel [28] (cf. Figure 1(a)).

Concept	Source MM			Target MM		
	Artifact	Columns	Class	Property	Associations	OCL constraints
<i>(a) A Project is mapped onto a Process</i>						
#Project	Msp_Projects	proj_uid name	Process Activity	oid_Process name	is_a Activity	
Project name						
<i>(b) Project Tasks are mapped onto Process Activities</i>						
#Activity		task_uid	Activity	oid_Activity	+activities (*)	
Activity name	Msp_Tasks + FK_Msp_Projects	task_name		Name	is_a Task	
Activity start		task_act_start		Start	is_a Milestone	
Activity end		task_act_end		End		
<i>(c) Group Activities are mapped onto Subprocesses</i>						
Activity is composed of a group of tasks	Msp_Tasks: Table + FK_Task_Parent	Parent of the set_of_rows with the same foreign key (Task_Parent_Uid = Task_uid) and (Msp_Projects.Uid = Msp_Tasks.Proj_Uid)	Activity Subprocess	Is_AdHoc = true	is_a Subprocess is_a Process	
<i>(d) Subprojects are mapped onto Subprocesses</i>						
Activity is an external subproject	Msp_Tasks: Table + FK_Task_Is_SubProject	Parent of the set_of_rows with the same foreign key Task_Is_SubProj_Proj_Uid = Proj_uid	Activity Subprocess		is_a Subprocess is_a Process	
<i>(e) Allowed Durations are mapped onto OCL constraints</i>						
Fixed duration	Msp_Tasks: Table	task_dur_is_est = false	Time_Constraint	mindur = maxdur	tc(*)	FIXD
Flexible duration		task_dur_is_est = true		mindur = task_dur		FLEXD
<i>(f) Time Constraints are mapped over Activity events</i>						
Inflexible TC	Msp_Tasks: Table	task_dur_is_est = false	Time_Constraint	mindur = maxdur	tc(*)	FIXD
Flexible TC		task_dur_is_est = true		mindur = task_dur		FLEXD
<i>(g) Time Dependencies are mapped over Activity events</i>						
Time Dependencies	Msp_Links FK_Link_Type FK_Predecessor FK_Successor	Link_type link_lag	Time_Dependency	td_type = link_type leadORlag = link_lag	+predecessor +successor	SS, SF, FS, FF

defined in PIM Process Metamodel [28] as OCL invariants and derivation rules.

Algorithm 1 shows the procedure to carry out M2M transformations whose mapping details are shown in Table 4.

5. Generating Software Business Processes by Running the Roadmap from Source MS Project Software Plans: The AQUA-WS Project Case Study

AQUA-WS Project [29] is a multiyear software modernization project leading a new Web-based system by means of multiple and heterogeneous old client-server legacy systems (cf. Figure 6, which depicts four subsystems composed of sixteen applications for water lifecycle management and infrastructures).

The project has been carried out for EMASESA, which is a public company in the utility sector that runs the water lifecycle in the city of Seville. It has been developed by international software companies in liaison with some research groups, from the University of Seville and the University of Malaga, amounting an investment of 3.5 million euros. Our research group, Web Engineering and Early Testing (<http://iwt2.org>), has been responsible for the work in methodological support and quality assurance of the project. We want to highlight that we have chosen this case not for the management of the water cycle that is the business of EMASESA but for the management of the software development process that has involved all the actors in this significant project.

The AQUA-WS case study [29] has allowed us to validate the proposed MDA-based framework. It has also helped us to extend the proposal to generate and merge processes from different LISs including more perspectives, such as *Organizational*, *Resources*, *Data*, and *Case* dimensions. In light of this, we describe the environment of the study in Section 5.1.

5.1. Environment of the Study. All the teams have used either stand-alone clients or Web UI interfaces of MPS® to face up their work. This software collects and centralizes all the information related to the development of different subsystems in those cases where the responsibility is shared among teams. AQUA-WS is just a case of the MDA-based initial roadmap developed in Section 4.2.

NDT methodology [31] has been the reference used to manage the AQUA-WS Project, and NDTQ-Framework [30] has offered facilities to support NDT and automatically generate documentation in an MDE-based project. NDTQ-Framework is implemented with Sparx Systems Enterprise Architect, which has helped us to customize M2M transformations by means of OMG Query/View/Transformation (QVT) [65] language and plugins. The project has been organized with different teams: (i) development team of each software company; (ii) quality assurance team; and (iii) customer team.

5.2. Analysis of Results. In this section, we analyze results obtained by means of the approach to generate software business processes from MPS® AQUA-WS legacy database, as the source LDB. Table 5 summarizes the transformations from units of the source project plan into target process elements. Figure 7 is the source Gantt chart regarding the schedule of software development *Activities* of AQUA-WS project [29], and Figures 8–10 represent processes generated by the MDA-based approach.

Processes are depicted as BPMN diagrams although it is easy to show them in other languages selected by software experts, such as SPEM, which is more appropriate to deal with this type of business processes. It is easy to change the final step of the reverse engineering procedure, which transform our PIM Metamodel [28] into the software process modeling language. This is because our PIM Metamodel [28] has a minimum number of classes and associations that always exist in metamodels of these software process modeling languages. Particularly, the transformation to the BPMN Metamodel also allows us to export results to other metamodels of software process modeling languages by using XML standard serialization formats such as *.xsd* and *.xmi*. These standards facilitate the interoperability between technological environments to exchange the obtained processes.

We have gathered three kinds of *activities* to apply the proposed MDA-based process generation (cf. Figure 2, which depicts most of the possible roadmaps and Figure 4, which illustrates the specific roadmap that is solved in this paper). It uses metamodels (Figures 1(a) and 4) and mapping rules (Algorithm 1 and Table 4) that we have previously shown. Each category of *activities* and related processes are described as follows:

- (i) *Organization, Quality Assurance, and Subsystems Decomposition.* This level is a composition of general *Activities* either for organizational purpose or for specific quality assurance work (cf. Figure 8), together with a decomposition into AQUA-WS subsystems (cf. Figure 6), where each subsystem is allocated to the main development team.
- (ii) *Development of Subsystems.* We have selected a subsystem, such as “Activity #61 (Alfa 0.4) Customers: Networks Intervention Subsystem” (cf. Figure 9: “Alfa 0.4. Customers. Networks Intervention Subsystem”) to display its generated business process. Subsystems involve processes linked to some *Activities* included in NDT waterfall software lifecycle [31] (cf. Figure 11), from the requirement to implementation phases.
- (iii) *NDT Phases.* Each NDT phase consists of *Activities* that may be optional or mandatory. For example, Figure 12 represents the corresponding process for the SA NDT phase, which, in this case, has been manually designed by the business software expert. Project manager uses patterns of NDT phases that are stored within the MPS® database. For instance, “Activity 1001” is an

```

Input:  $J$  project, which is stored in MPS® Database Published instance
Output: BP Process, which conforms to platform-independent (PIM) Process Metamodel [28] that includes OCL Time Rules
BP  $\leftarrow$  new(Process); —They refer to mappings included in Table 4(a)
for each ((TMP: Msp_tasks)  $\in$  (J: Msp_Projects)){ —It maps MPS® Tasks onto PIM MM Activities
  A  $\leftarrow$  create_Activity (TMP); —It creates A: new Activity  $\in$  BP for each project task
  create_TCs (A, TMP); —It creates TCs: duration, fixed or flexible events
  create_TDs (A, TMP); —It creates TDs: dependencies between current task and their predecessors
};
return BP;
function create_Activity (TMP: Msp_Task) { —It creates an Activity and its corresponding subclasses
  A  $\leftarrow$  new(Activity)  $\in$  BP; —The activity is included into BP (mappings included in Table 4(b))
  case (TMP) {
    “Subproject” or “Task_Group”: {
      P  $\leftarrow$  new(Process)  $\triangleright$  A; —Subprojects/Groups are mapped as subprocesses
      SP  $\leftarrow$  new(Subprocess)  $\triangleright$  P; SP  $\triangleright$  A; —It sets the hierarchy of Activities and subprocesses (Table 4(d))
      if (TMP == “Task_Group”) SP.isAdHoc := true; —It groups tasks as AdHoc subprocesses (Table 4(c))
    }
    “Single Task”: T  $\leftarrow$  new(Task)  $\triangleright$  A; —Activity is a single task
    “Milestone”: M  $\leftarrow$  new(Milestone)  $\triangleright$  A; —It is a milestone
  }
};
function create_TCs (A: Activity, TMP: Msp_Task) { —It creates TCs: duration and time events
  TC  $\leftarrow$  new(Time_Constraint)  $\in$  A; —It creates TC for activity duration (Table 4(e))
  if (TMP.task_dur_is_est) then {TC.tc_type  $\leftarrow$  “FLEXD”; —It refers to estimated duration as Flexible Duration}
  else {TC.tc_type  $\leftarrow$  “FIXD”; —It is fixed duration};
  if (TMP.task_constraint_type  $\in$  {“MSON”, “SASAP”, “SALAP”, “SNET”, “SNLT”, “MFON”, “FASAP”, “FALAP”, “FNET”, “FNLT”})
  then {
    TC  $\leftarrow$  new(Time_Constraint)  $\in$  A; —It creates TC for scheduled end events (Table 4(f))
  }
};
function create_TDs (A: Activity, TMP: Msp_Task) { —It creates TDs {“SS”, “SF”, “FS”, “FF”}
  for each (LK: Msp_links)  $\in$  (TMP: Msp_Task) { —It maps links onto dependencies
    TD  $\leftarrow$  new(Time_Dependency)  $\in$  A; —It creates TD for scheduled events (Table 4(g))
  }
};

```

ALGORITHM 1: PSM_MSProjectDatabase_TO_PIM_process_MM (J : Msp_Projects).

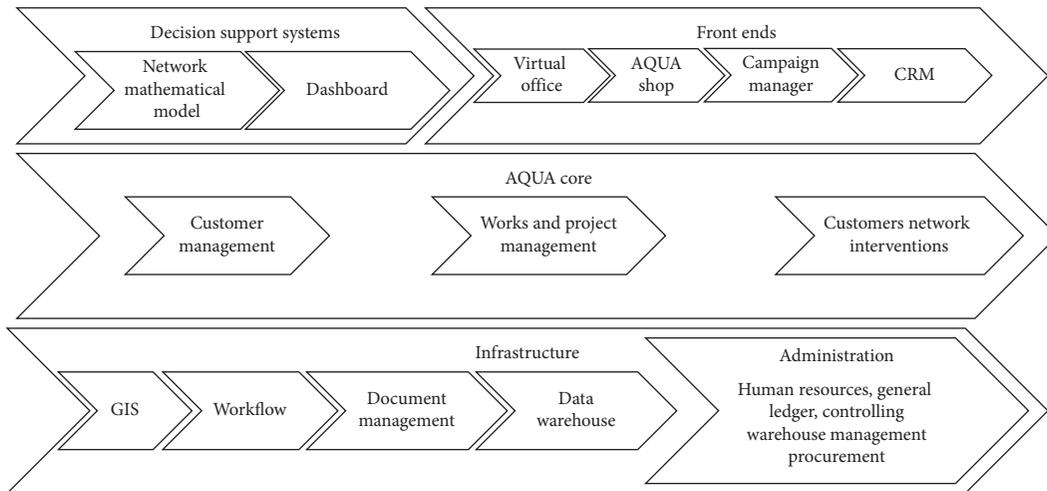


FIGURE 6: Information System Architecture of Emasesa, which is the object of AQUA-WS software development project; subsystems and applications to manage the cycle of water supply and sanitation networks.

TABLE 5: Mappings from source “Software Project Plans” onto target “Software Business Processes”.

Source scheduled tasks		Target process elements	
Group of tasks	No. of tasks (Figure 7)	Process activities generated	Figures
Organizational Activities and Subsystems	#(1, 3) #(11, 40, 56, 61, 80, 94)	Planning and control together with quality assurance activities; parallel development of subsystems.	Figure 8
Decomposition of global task (Alfa 0.4) Customers: Networks Intervention into detailed tasks	#61 decomposed into #(62–69)	Process that represents a decomposition of a software subsystem into development phases. Phases that concern NDT methodology, which is applied in the context of software waterfall development lifecycle (Figure 11). Decomposition of a concrete SA NDT phase associated with the application included in the subsystem #61 and related to SA NDT phase. Four models are generated; therefore,	Figure 9
External subproject activities linked to a System Analysis (SA) NDT phase	#64	they may be compared to the NDT SA pattern phase (Figure 12).	Figure 10

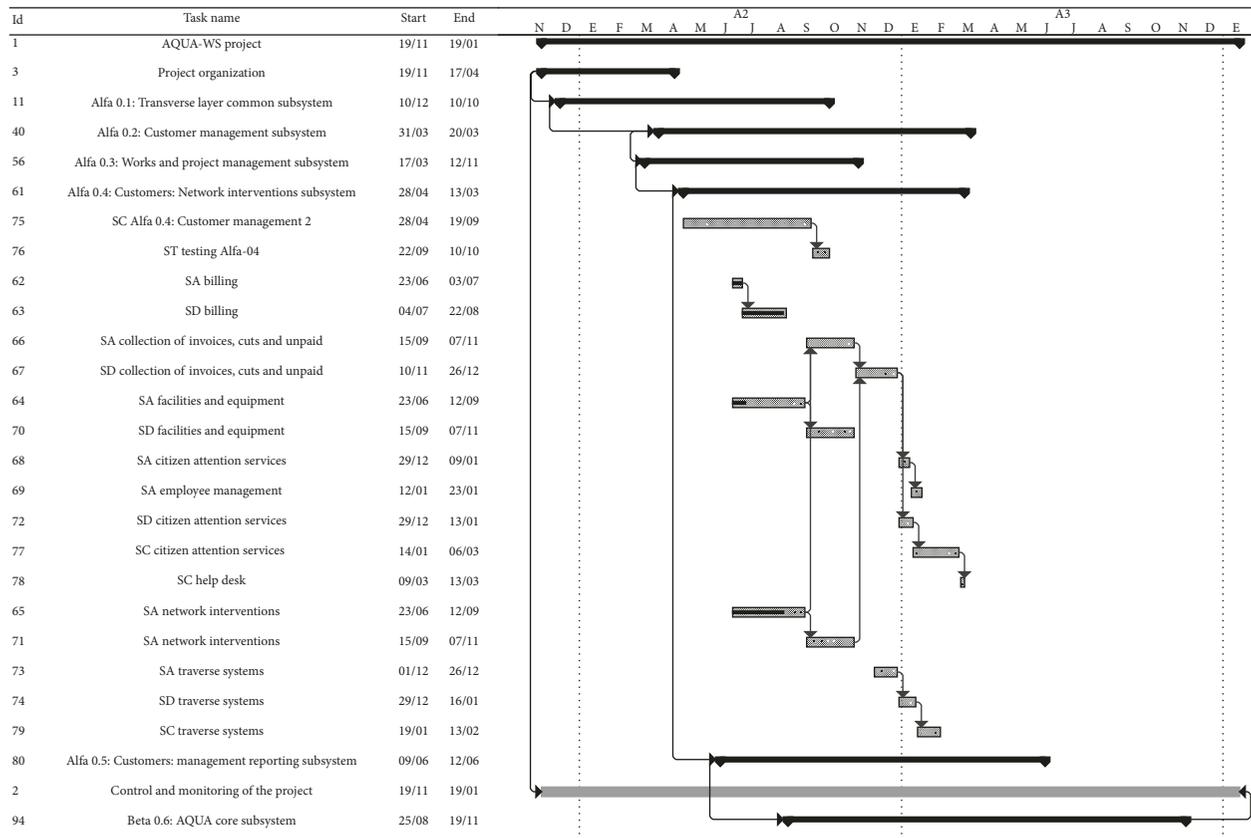


FIGURE 7: AQUA-WS project schedule. Organization and quality assurance activities, subsystems scheduling, and detailed NDT phases of subsystem “Alfa 0.4”.

external subproject for NDT SA phase that may be linked to a specific SA Activity contained in a subsystem. Later, Activities inherited from the pattern may be modified in their corresponding subsystems. We have chosen “Activity #64 SA of Facilities and Equipment Subsystem” (cf. Figure 10) in this third category.

We comment on some aspects of process generation as follows:

- (i) To start, *Project Tasks* are mapped onto *Process Activities* and *Control Flow* in each piece of the process appears as a consequence of *TDs*. On the one hand, *FS TD* induces sequential flow (\diamond); on the other hand, *SS, SF, and FF TDs* are represented with parallel flows (\diamond), without additional restrictions.
- (ii) Next, *Hierarchies of Activities* are solved as follows:
 - (a) *Parent Activity* is mapped onto an *AdHoc Subprocess*; (b) *Child Activities* run in parallel.

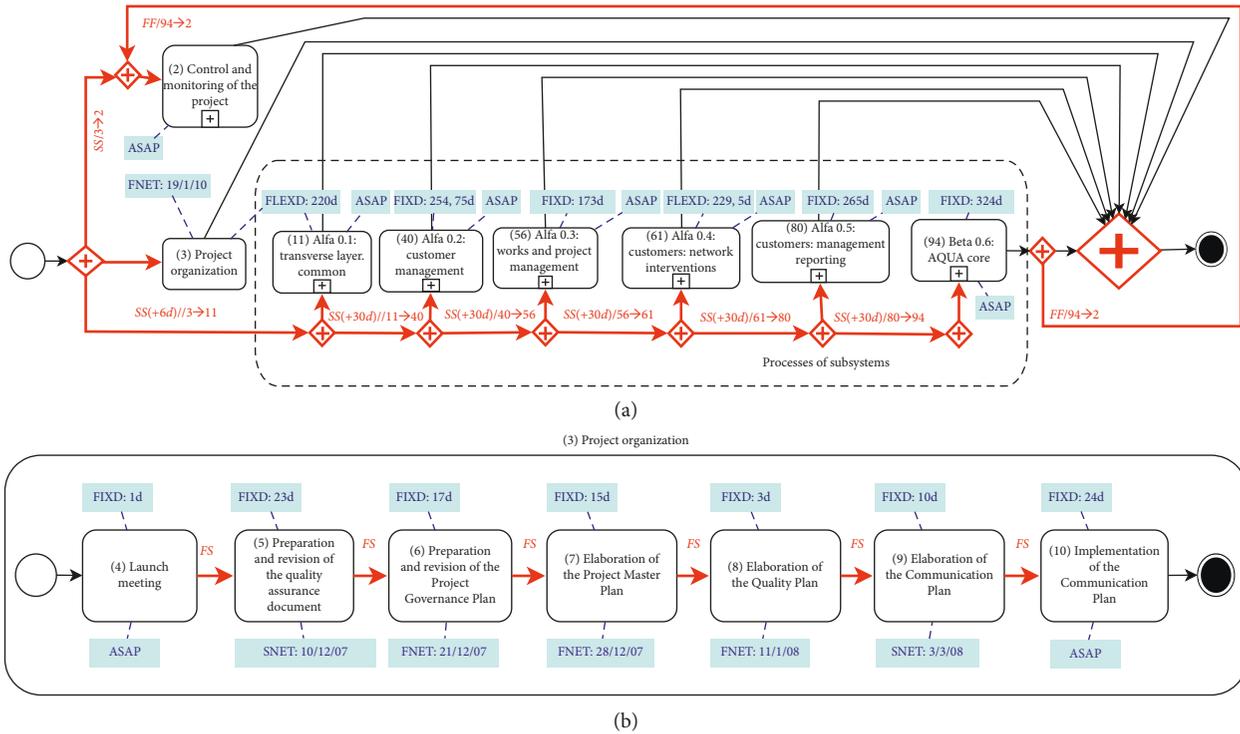


FIGURE 8: Generated process for MPS® “Organizational Activities and Subsystems”. Control Flow of activities and Time Rules (TCs and TDs) extracted from the project plan.

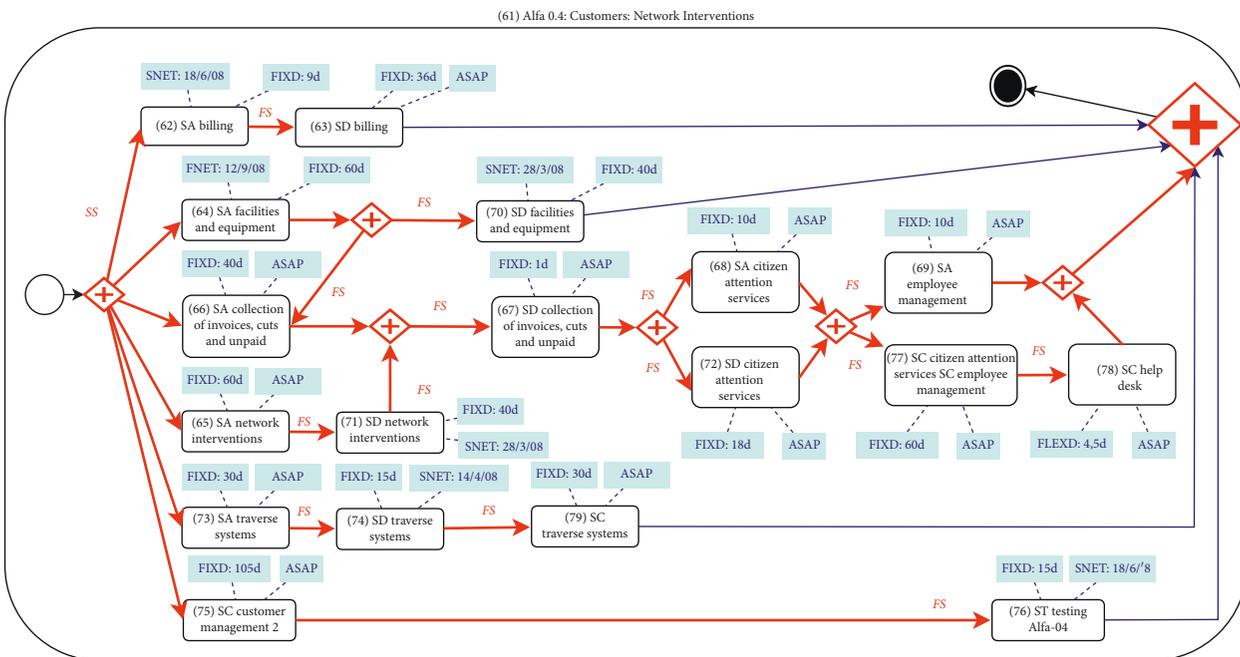


FIGURE 9: Generated process: “Activity #61 (Alfa 0.4) Customers: Networks Intervention Subsystem.” Control Flow of Activities and Time Rules (TCs and TDs) extracted from the project plan.

Attending to time semantics generated processes enforce the selected *Time Rules* of our Metamodel [28] (cf. Figure 1(a)), which are expressed by OCL assertions [28] through our MDA-based approach.

(iii) To finish, *TCs* are allocated to their corresponding *Activities*, enriching each generated process. We can assure that all tasks in a project have been properly captured and grouped into *Subprocesses*

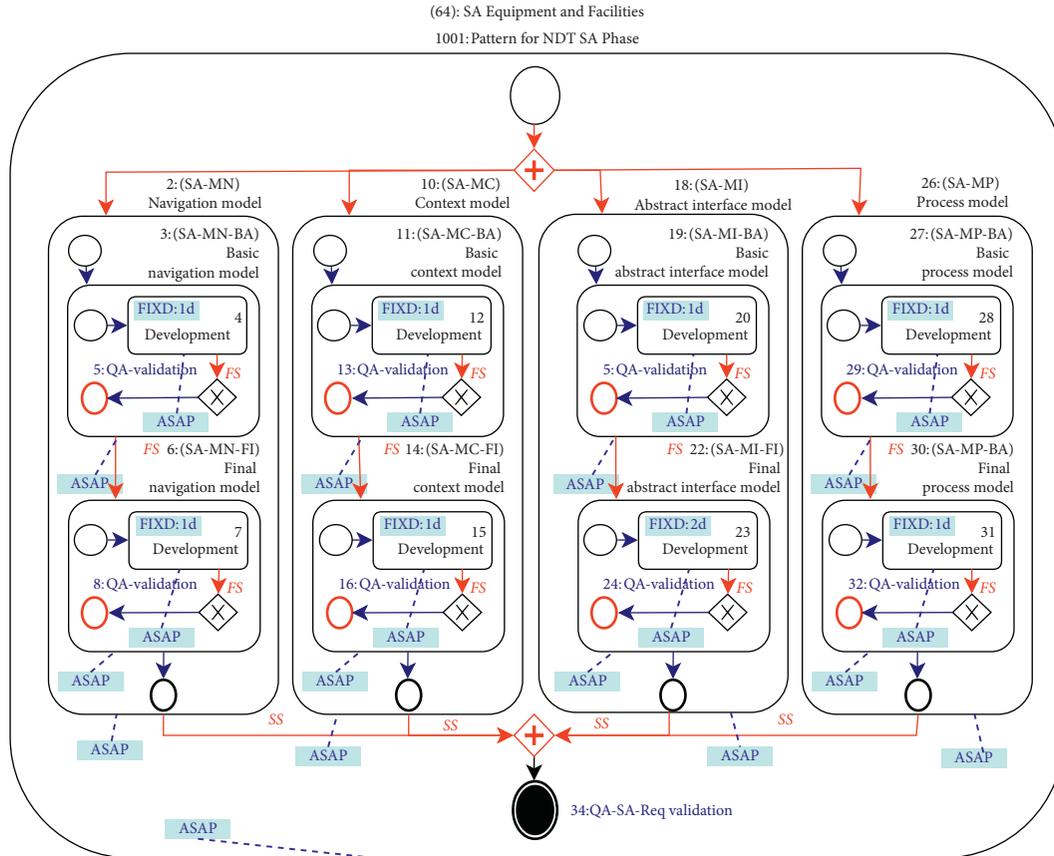


FIGURE 10: Generated BP: “Activity #64 SA of Facilities and Equipment.” Control Flow of Activities and Time Rules (TCs and TDs) extracted from the project plan.

and *Processes* by means of *Time Dimension* identification as base heuristics.

We have compared reference NDT software business processes [31], manually designed by business experts, with processes generated by our MDA-based approach (such as processes corresponding to Figure 9 vs Figure 11 and Figure 10 vs Figure 12). It is worth mentioning that the generated processes are useful for business software experts and constitute a good approximation to those real that could be designed manually. Nevertheless, we can observe that the absence of advanced constructions (such as loops or transactions) in the generated processes is due to limitations of the source system (MPS®), which neither allows iterations over links nor more powerful logic rules for grouping tasks.

As the level of abstraction is concerned, we have been generating process instances from project plans, although sometimes *Activities* may be allocated to an *Activity* category in the scope of NDT reference [31] (cf. the use of patterns above). Therefore, our approach may generate process instances that are M_0 models regarding the Meta Object Facility (MOF) concept [8]. It may also capture a higher level of abstraction, such as M_1 models, in those cases where heuristics are applied to patterns or individual *Activities* are allocated to an *Activity* category. Process Mining Algorithms work in the same way, which means to

know the task type of each executed, for instance, that is stored in the *event log*.

The approach could be easily applied to other contexts, such as (i) Project-Oriented Organizations, out of the business software sector, which also use MPS®; and (ii) New PMSs as sources, which are as widely used as MPS®, such as Redmine®. This way, the approach may be monetized running on a large number of projects carried out by many organizations.

Challenges to suite this approach to other PMSs are as simple as extracting a new PMS task Metamodel (as shown in Figure 4) and rewriting M2M mappings (Algorithm 1 and Table 4) from the new source by delivering a new roadmap (Figure 2). Business Processes may be enriched with other perspectives, such as *Organizational*, *Resources*, and *Costs*, which may be found in PMSs. Other Non-PAISs, which support the software lifecycle, could also play the role of sources (such as ECMs, ERPs, CRMs, SCMs, or Tailor-Made software). If multiple Non-PAISs were used in the same organization, then fragments of evidence regarding the same business process could be split into different databases. As a result, the approach would need to be strengthened in order to consistently merge process fragments into a unique conceptual process.

Generated processes with this approach are instances that we could consider raw results, which a human expert would have to review; however, we could consider the

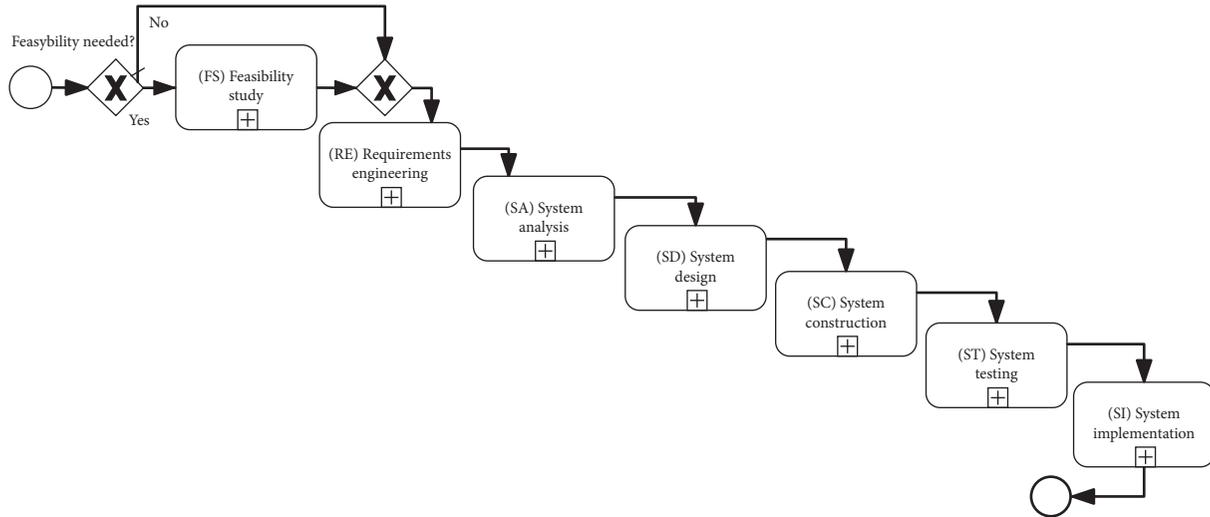


FIGURE 11: Navigational development technique (NDT). Waterfall software lifecycle.

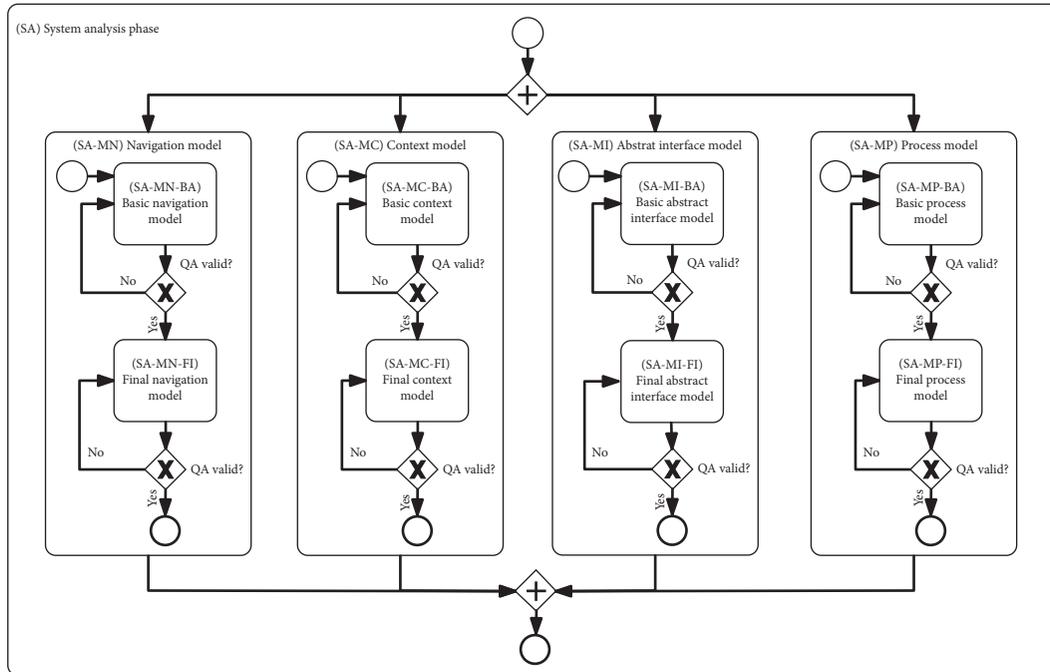


FIGURE 12: SA NDT phase manually modeled by an expert software engineer in NDT.

possibility of enriching the transformation algorithms with process refactoring patterns to improve the quality of them regarding parameters such as comprehensibility and modifiability [40–42].

6. Conclusion and Future Work

Software experts are increasingly using the BPM approach, making ITCs become more competitive in today’s globalized world. Process Mining Techniques [17–19], as the major exponent of BPD, are well suited to obtain processes from PAIS *event logs*, but not from Non-PAISs because they lack these artifacts. Some Non-PAISs used by ITCs may hide a lot of knowledge about the execution of software business

processes. Specifically, PMSs, such as MPS®, widely used by ITCs, is a source of project plans that may be transformed into processes, since they offer more advantages (i.e., BPM lifecycle of continuous improvement [2]) than just planning projects.

Regarding BPD from Non-PAISs, in which Process Mining is not applicable, we propose an MDA-based framework allowing different roadmaps to transform database artifacts, regarding tasks of ITCs, into processes of ITCs. Then, the proposal is useful to apply BPD to Non-PAISs of ITCs, such as PMSs. Based on this framework, we have developed a specific MDA-based roadmap (from MPS® *database* to *Process Metamodel* [28]) to convert project plans, stored in an MPS® database, into software business

processes, which conform to *Process Metamodel* [28]. That allows interoperability with BPMLs (SPMLs or GPMLs) used in the software field. The challenges that arise with this approach are (i) to use LISs for software lifecycle management and (ii) to study and define the database metamodel of tasks that allows exploring the hidden dimensions (initially *Control flow* and *Time Dimension*, but extendable to others) of processes.

The related works we have further analyzed let extract processes from Non-PAISs by using heuristics or tools like SDG [36], ASTM [10, 23–27], and KDM [10, 23, 24]. Unlike our approach, they need a lot of extra effort to approximate to the reality of processes, so they have not been sufficiently applied to industrial contexts. To the best of our knowledge, we have not found out any study such as our MDA-based approach, whose main characteristics are as follows:

- (i) It is initially focused on ITCs and PMSs as source systems since business processes are organized as projects whose states are stored in databases of PMSs.
- (ii) Heuristics apply M2M transformations based on *Time Dimension* of source projects, which are in correspondence with processes defined with target *Process Metamodel* [28] (cf. Figure 1(a)); meanwhile, related approaches only use ASTM and KDM. Metamodeling-based reverse engineering procedure provides a high level of interoperability to our approach.
- (iii) Results are processes close to business software expert, which may be defined with BPMLs. If the target BPML allows standard XML exchange formats, such as .xsd, .xmi, or .xpdl, then, results will be available to other roadmaps with the same effort. Later, the business expert must analyze and complete processes in a BPM lifecycle of continuous improvement [2].

To summarize, our proposal provides a framework to generate software business processes that would otherwise be hidden or wasted in databases of Non-PAISs. This hidden knowledge can be used to implement the BPM approach in ITCs that will help them to become more competitive and reduce costs. Compared to other BPD methods [10, 23–27, 36] used with Non-PAISs, our results are more adjusted to the reality of processes since we focus on transformations among artifacts that are close to executed processes that exist at different levels of abstraction (i.e., platform level and software expert level). Furthermore, business processes may be enriched with data regarding *Resources* and *Costs* that may also be bound to projects in PMSs. This way, new data will be available to set metrics and study Key Performance Indicators (KPIs) of software business processes.

This paper illustrates the AQUA-WS project case study [29] to test the developed MDA-based roadmap (i.e., from MPS® database to *Process Metamodel* [28]). In this case study, we have shown that generated processes are similar to real processes that a business software expert may design.

For this reason, we have delivered a semiautomatic proposal to obtain processes of ITCs.

We acknowledge that this study is only a first step towards validating the approach. With regard to future work,

- (i) First, we plan to validate the approach with more cases following the steps:
 - (a) Applying the proposal to more MPS® case studies with other software lifecycles, such as methodologies based on Agile Methods [66, 67]. The developed MPS® roadmap is reusable in many cases, thus the new inputs will be MPS® plans involving categorized activities concerning software development projects. Activities must be classified in a category in the same way that Process Mining Techniques [17–19] use traces stored in PAISs *event logs*, which means that it is necessary to know the type of each particular task. We are working in a prototype to reverse engineering MPS® databases of ITCs, which is based on Enterprise Architect (<http://www.sparxsystems.com>) customization. We are trying to organize the exploitation of this approach in the Andalusian government bodies (cf. the EMPOWER [68] project).
 - (b) Testing the same approach either with other PMSs, such as Redmine®, with more types of Non-PAISs, such as ECMs (e.g., Alfresco® or SharePoint®), popular ERPs (licensed like SAP®, Oracle or Microsoft, or open sources like Open Bravo® or Odoo®) or Tailor-Made Software, among others.
 - (c) Evaluating the approach out of the IT field, which means in other project-oriented industrial sectors that utilize PMSs.
 - (d) Based on a set of significant cases, performing more solid statistical validation of the proposal to measure efficiency and effectiveness indicators of our approach for the automatic generation of software business processes in comparison with manual methods that an expert in this field could use.
- (ii) Second, another line of research should cope with extending the *Process Metamodel* [28] (c.f. Figure 1(a)) to involve other process dimensions such as *Resource*, *Case*, and *Data*. As *Time Dimension* is concerned, the affinity of our Metamodel [28] with metamodels proposed by Awad et al. [69] and Stroppi et al. [70], related to *Resource* perspective, could enrich target process models including *Time* and *Resource* dimensions. In this case, the proposal will need to be extended to allow merging process fragments from different source legacy databases used by the same organization.
- (iii) Third, generated processes are raw results that may lack a certain quality with respect to comprehensibility and modifiability. In this sense, we could

consider enriching the transformation algorithms by means of refactoring techniques [40–42] so that human-perceived quality measures could be improved.

- (iv) Finally, we aim to generate standard *event logs*, such as XES format [19], for Non-PAISs. It should help to combine our approach with Process Mining Techniques [17–19] in order to compare proposed processes.

Data Availability

The AQUA-WS data used to support the findings of this study are available upon request to the authors.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

C. Arevalo provided the initial idea and guided the whole process of the manuscript. I. Ramos has contributed adapting meta-models of processes for the extraction of temporary rules from legacy systems. J. Gutiérrez has contributed to the creation of the general approach based on MDE for the generation of software processes from legacy systems used by ITCs. M. Cruz helped design the AQUA-WS case study, execute this specific roadmap to generate software processes, complete data analysis, and compare automatically generated processes with manual processes developed by ITC software experts.

Acknowledgments

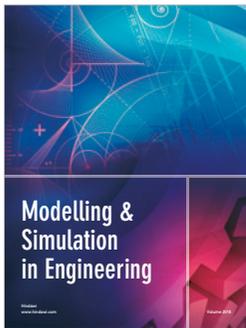
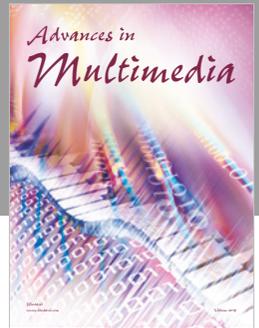
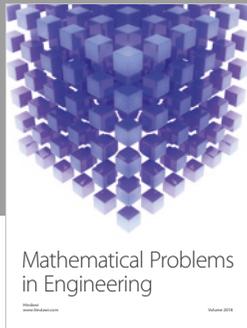
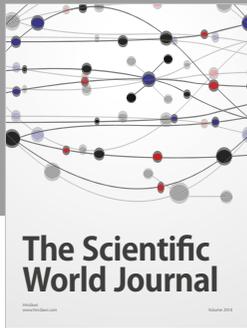
This research has been supported by the POLOLAS project (TIN2016-76956-C3-2-R) of the Spanish Ministry of Economy and Competitiveness.

References

- [1] W. M. van der Aalst, "Business process management: a comprehensive survey," *ISRN Software Engineering*, vol. 2013, Article ID 507984, 37 pages, 2013.
- [2] M. Dumas, W. M. P. van der Aalst, and A. H. M. ter Hofstede, *Process-Aware Information Systems: Bridging People and Software through Process Technology*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2005.
- [3] J. Luftman, H. S. Zadeh, B. Derksen, M. Santana, E. H. Rigoni, and Z. Huang, "Key information technology and management issues 2012-2013: an international study," *Journal of Information Technology*, vol. 28, no. 4, pp. 354–366, 2013.
- [4] O. Henfridsson, L. Mathiassen, and F. Svahn, "Managing technological change in the digital age: the role of architectural frames," *Journal of Information Technology*, vol. 29, no. 1, pp. 27–43, 2014.
- [5] S. Beecham, T. Hall, and A. Rainer, "Software process improvement problems in twelve software companies: an empirical analysis," *Empirical Software Engineering*, vol. 8, no. 1, pp. 7–42, 2003.
- [6] F. Ruiz-González and G. Canfora, "Software process: characteristics, technology and environments, UPGrade," *The European Journal for the Informatics Professional*, vol. 5, no. 5, pp. 6–10, 2004, <http://www.cepis.org/upgrade/files/full-2004-V.pdf>.
- [7] S. Kent, "Model driven engineering," in *Proceedings of the Third International Conference on Integrated Formal Methods*, pp. 286–298, Turku, Finland, May 2002.
- [8] A. Kleppe, J. Warmer, and W. Bast, *MDA Explained: The Model Driven Architecture: Practice and Promise*, Addison-Wesley Professional, Boston, MA, USA, 2003, ISBN: 032119442X.
- [9] P. Mohagheghi, W. Gilani, A. Stefanescu, and M. A. Fernandez, "An empirical study of the state of the practice and acceptance of model-driven engineering in four industrial cases," *Empirical Software Engineering*, vol. 18, no. 1, pp. 89–116, 2013.
- [10] R. Pérez-Castillo, I. García-Rodríguez de Guzmán, M. Piattini, and Á. S. Places, "A case study on business process recovery using an e-government system," *Software: Practice and Experience*, vol. 42, no. 2, pp. 159–189, 2012.
- [11] B. Gedik and H. Andrade, "A model-based framework for building extensible, high performance stream processing middleware and programming language for IBM InfoSphere Streams," *Software: Practice and Experience*, vol. 42, no. 11, pp. 1363–1391, 2012.
- [12] X. Zhao and Y. Zou, "A business process-driven approach for generating software modules," *Software: Practice and Experience*, vol. 41, no. 10, 2011.
- [13] J. Bisbal, D. Lawless, B. Wu, and J. Grimson, "Legacy Information Systems: Issues and Directions," *IEEE Software*, vol. 16, no. 5, pp. 103–111, 1999.
- [14] W. M. Ulrich, *Legacy Systems: Transformation Strategies*, Prentice Hall, Englewood Cliffs, NJ, USA, 2002, ISBN 013044927X, <http://dl.acm.org/citation.cfm?id=515375>.
- [15] R. C. Seacord, D. Plakosh, and G. A. Lewis, *Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices*, Addison-Wesley, Boston, MA, USA, 2003, ISBN 0321118847, <http://dl.acm.org/citation.cfm?id=599767>.
- [16] A. De Lucia, R. Francese, G. Scanniello, and G. Tortora, "Developing legacy system migration methods and tools for technology transfer," *Software: Practice and Experience*, vol. 38, no. 13, pp. 1333–1364, 2008.
- [17] W. M. P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, Springer, Berlin, Germany, 2011.
- [18] W. M. P. Van der Aalst, "A general divide and conquer approach for process mining," in *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS 2013)*, pp. 1–10, Kraków, Poland, September 2013, ISBN 9781467344715, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6643968>.
- [19] H. M. W. Verbeek, J. C. A. M. Buijs, B. F. Van Dongen, and W. M. P. van der Aalst, "XES, XESame, and ProM 6," in *Proceedings of the International Conference on Advanced Information Systems Engineering (CAiSE Forum 2010)*, pp. 60–75, Hammamet, Tunisia, June 2011.
- [20] W. M. P. van der Aalst, "Extracting event data from databases to unleash process mining," in *Management for Professionals*, pp. 105–128, Springer International Publishing, Berlin, Germany, 2015.
- [21] S. Adam, N. Riegel, J. Doerr, O. Uenalan, and D. Kerkow, "From business processes to software services and vice versa-

- an improved transition through service-oriented requirements engineering,” *Journal of Software: Evolution and Process*, vol. 24, no. 3, pp. 237–258, 2012.
- [22] Y. Zou, J. Guo, K. C. Foo, and M. Hung, “Recovering business processes from business applications,” *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 21, no. 5, pp. 315–348, 2009.
- [23] R. Pérez-Castillo, M. Fernández-Ropero, I. G. R. de Guzmán, and M. Piattini, “MARBLE. A business process archeology tool,” in *Proceedings of the 2011 27th IEEE International Conference on Software Maintenance (ICSM)*, pp. 578–581, IEEE, Williamsburg, VA, USA, September 2011.
- [24] R. Pérez-Castillo, I. G.-R. de Guzmán, and M. Piattini, “Business process archeology using MARBLE,” *Information and Software Technology*, vol. 53, no. 10, pp. 1023–1044, 2011.
- [25] C. Arevalo, *Una prop uesta basada en el paradigma dirigido por modelos para la extracción de procesos del software desde sistemas heredados utilizando la perspectiva temporal*, Ph.D. thesis, University of Seville, Seville, Spain, 2016, <https://idus.us.es/xmlui/handle/11441/42996>.
- [26] C. Arevalo, M. T. Gómez-López, A. M. Reina Quintero, and I. Ramos, “An architecture to infer business rules from event condition action rules implemented in the persistence layer,” in *Uncovering Essential Software Artifacts through Business Process Archeology*, R. Perez-Castillo and M. Piattini, Eds., Business Science Reference, Hershey, PA, USA, pp. 201–221, 2013.
- [27] C. Arevalo, I. Ramos, and M. J. Escalona, “Discovering business models for software process management—an approach for integrating time and resource perspectives from legacy information systems,” in *Proceedings of the 17th International Conference on Enterprise Information Systems (ICEIS 2015)*, pp. 353–359, SCITEPRESS - Science and Technology Publications, Barcelona, Spain, April 2015.
- [28] C. Arevalo, M. J. Escalona, I. Ramos, and M. Domínguez-Muñoz, “A metamodel to integrate business processes time perspective in BPMN 2.0,” *Information and Software Technology*, vol. 77, pp. 17–33, 2016.
- [29] C. R. Cutilla, J. A. García-García, J. J. Gutiérrez et al., “Model-driven test engineering: a practical analysis in the AQUA-WS project,” in *Proceedings of the 7th International Conference on Software Paradigm Trends (ICSOFT 2012)*, pp. 111–119, Rome, Italy, July 2012, ISBN:9789898565198, <http://www.scopus.com/inward/record.url?eid=2-s2.0-84868673216&partnerID=tZOtx3y1>.
- [30] J. Ponce, L. García-Borgoñón, J. A. García-García et al., “A model-driven approach for business process management,” *Covenant Journal of Engineering & Technology (CJICT)*, vol. 1, no. 2, pp. 32–52, 2013, <http://journals.covenantuniversity.edu.ng/index.php/cjict/article/view/79>.
- [31] M. J. Escalona, M. Mejías, J. Torres, and A. Reina, “The NDT development process,” in *Proceedings of the International Conference on Web Engineering*, pp. 463–467, Springer, Oviedo, Spain, July 2003.
- [32] P. Newcomb, “Architecture-driven modernization (ADM),” in *Proceedings of the 12th Working Conference on Reverse Engineering (WCRE’05)*, p. 237, IEEE, Pittsburgh, PA, USA, November 2005.
- [33] R. Pérez-Castillo, I. García-Rodríguez de Guzmán, I. Caballero, and M. Piattini, “Software modernization by recovering web services from legacy databases,” *Journal of Software: Evolution and Process*, vol. 25, no. 5, pp. 507–533, 2013.
- [34] E. J. Chikofsky and J. H. Cross, “Reverse engineering and design recovery: a Taxonomy,” *IEEE Software*, vol. 7, no. 1, pp. 13–17, 1990.
- [35] E. J. Byrne, “Software reverse engineering: a case study,” *Software: Practice and Experience*, vol. 21, no. 12, pp. 1349–1364, 1991.
- [36] A. Cleve, J. Henrard, and J. Hainaut, “Data reverse engineering using system dependency graphs,” in *Proceedings of the 2006 13th Working Conference on Reverse Engineering*, pp. 157–166, IEEE, Benevento, Italy, October 2006.
- [37] OMG, “Unified modeling language (UML®),” 2015, <http://www.omg.org/spec/UML/>.
- [38] V. Cosentino and S. Martinez, “Extracting UML/OCL integrity constraints and derived types from relational databases,” in *Proceedings of the 13th International Workshop on OCL, Model Constraint and Query Languages*, pp. 43–52, Miami, FL, USA, September 2013, <http://hal.univ-nantes.fr/hal-00869231/>.
- [39] M. Zanoni, F. Perin, F. A. Fontana, and G. Viscusi, “Pattern detection for conceptual schema recovery in data-intensive systems,” *Journal of Software: Evolution and Process*, vol. 26, no. 12, pp. 1172–1192, 2014.
- [40] D. Caivano, M. Fernández-Ropero, R. Pérez-Castillo, M. Piattini, and M. Scalera, “Artifact-based vs. human-perceived understandability and modifiability of refactored business processes: an experiment,” *Journal of Systems and Software*, vol. 144, pp. 143–164, 2018.
- [41] R. Pérez-Castillo, M. Fernández-Ropero, M. Piattini, and D. Caivano, “How does refactoring affect understandability of business process models?,” in *Proceedings of the 25th International Conference on Software Engineering & Knowledge Engineering (SEKE’13)*, pp. 644–649, Knowledge Systems Institute Graduate School, Boston, MA, USA, June 2013, ISBN-13: 978-1-891706-33-2.
- [42] R. Pérez-Castillo, M. Fernández-Ropero, and M. Piattini, “Business process model refactoring applying IBUPROFEN. An industrial evaluation,” *Journal of Systems and Software*, vol. 147, pp. 86–103, 2019.
- [43] L. García-Borgoñón, M. A. Barcelona, J. A. García-García, M. Alba, and M. J. Escalona, “Software process modeling languages: a systematic literature review,” *Information and Software Technology*, vol. 56, no. 2, pp. 103–116, 2014.
- [44] C. Portela, A. Vasconcelos, A. Silva et al., “A comparative analysis between BPMN and SPEM modeling standards in the software processes context,” *Journal of Software Engineering and Applications*, vol. 5, no. 5, pp. 330–339, 2012.
- [45] R. M. Pillat, T. C. Oliveira, P. S. C. Alencar, and D. D. Cowan, “BPMNt: a BPMN extension for specifying software process tailoring,” *Information and Software Technology*, vol. 57, pp. 95–115, 2015.
- [46] ISO, *ISO/IEC 19510: OMG Business Process Model and Notation (BPMN)*, International Organization for Standardization, Geneva, Switzerland, 2013.
- [47] OMG, “SPEM, software & systems process engineering metamodel specification 2.0,” 2008, <http://www.omg.org/spec/SPEM/>.
- [48] R. Bendraou, C. Benoît, X. Xavier, and M.-P. Gervais, “Definition of an executable SPEM 2.0,” in *Proceedings of the Asia-Pacific Software Engineering Conference, APSEC*, pp. 390–397, Nagoya, Japan, December 2007.
- [49] I. Ruiz-Rube, J. M. Dodero, M. Palomo-Duarte, M. Ruiz, and D. Gawn, “Uses and applications of software & systems process engineering meta-model process models. A

- systematic mapping study,” *Journal of Software: Evolution and Process*, vol. 25, no. 9, pp. 999–1025, 2013.
- [50] ISO, *ISO/IEC 24744: Software Engineering—Metamodel for Development Methodologies*, International Organization for Standardization, Geneva, Switzerland, 2014.
- [51] OMG, “Essence—kernel and language for software engineering methods (essence),” 2015, <http://www.omg.org/spec/Essence/>.
- [52] I. Jacobson, P.-W. Ng, P. E. McMahon, I. Spence, and S. Lidman, “The essence of software engineering: the SEMAT kernel,” *Communications of the ACM*, vol. 55, no. 12, pp. 42–49, 2012.
- [53] F. Bonnet, G. Decker, L. Dugan, M. Kurz, and Z. Misiak, “Making BPMN a true lingua franca, BPM Trends,” 2014, <http://www.bptrends.com/bpt/wp-content/uploads/06-03-2014-ART-MakingBPM-a-TrueLinguaFranca-Zbigniew-Misiak-et-al.pdf>.
- [54] C. Flores and M. Sepúlveda, “Temporal specification of business processes through project planning tools,” in *Business Process Management Workshops*, pp. 85–96, Springer Berlin Heidelberg, Berlin, Germany, 2011.
- [55] D. Gagné and A. Trudel, “Time-BPMN,” in *Proceedings of the IEEE Conference on Commerce and Enterprise Computing*, pp. 361–367, Vienna, Austria, July 2009.
- [56] S. Cheikhrouhou, S. Kallel, N. Guermouche, and M. Jmaiel, “Toward a time-centric modeling of business processes in BPMN 2.0,” in *Proceedings of International Conference on Information Integration and Web-Based Applications & Services (IIWAS’13), Proceedings of the ACM International Conference Proceeding Series*, pp. 154–163, Vienna, Austria, 2013.
- [57] E. González López De Murillas, W. M. P. van der Aalst, and H. A. Reijers, “Process mining on databases. Unearthing historical data from redo logs,” in *Proceedings of the 3th International Conference on Business Process Management*, pp. 367–385, Innsbruck, Austria, August–September 2015.
- [58] E. González López de Murillas, H. A. Reijers, and W. M. P. van der Aalst, “Connecting databases with process mining: a meta model and toolset,” in *Proceedings of the International Conference on Business Process Modeling, Development and Support (BPMDS 2016)*, pp. 231–249, Ljubljana, Slovenia, June 2016.
- [59] S. Jablonski and C. Bussler, *Workflow Management: Modeling Concepts, Architecture and Implementation*, International Thomson Computer Press, Boston, MA, USA, 1996, <http://www.citeulike.org/group/6987/article/3401448>.
- [60] R. G. Ross, *Business Rules Concepts: Getting to the Point of Knowledge*, Business Rule Solutions, LLC, Houston, TX, USA, 4th edition, 2009, https://www.brsolutions.com/b_concepts.php.
- [61] G. Wagner, “Rule modeling and markup,” in *Reasoning Web*, pp. 251–274, Springer, Berlin, Germany, 2005.
- [62] R. Engel, W. Krathu, M. Pichler et al., “Analyzing inter-organizational business processes,” *Information Systems and e-Business Management*, vol. 14, no. 3, pp. 577–612, 2016.
- [63] C. Gonzalez-Perez and B. Henderson-Sellers, *Metamodelling for Software Engineering*, Wiley Publishing, Hoboken, NJ, USA, 2008, ISBN 0470030364, 9780470030363, <http://dl.acm.org/citation.cfm?id=1502365>.
- [64] J. F. Allen, “Maintaining knowledge about temporal intervals,” *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, 1983.
- [65] OMG, “QVT, Meta Object Facility (MOF) 2.0 query/view/transformation,” 2011, <http://www.omg.org/spec/QVT/1.1/>.
- [66] K. Schwaber, “Scrum development process, business object design and implementation,” 1997, http://link.springer.com/10.1007/978-1-4471-0947-1_11.
- [67] J. Bosch and P. M. Bosch-Sijtsema, “Introducing agile customer-centered development in a legacy software product line,” *Software: Practice and Experience*, vol. 41, no. 8, pp. 871–882, 2011.
- [68] J. A. Garcia-Garcia, J. G. Enriquez, L. Garcia-Borgonon, C. Arevalo, and E. Morillo, “A MDE-based framework to improve the process management: the EMPOWER project,” in *Proceedings of the 2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, pp. 553–558, IEEE, Emden, Germany, July 2017.
- [69] A. Awad, A. Grosskopf, A. Meyer, and M. Weske, *Enabling Resource Assignment Constraints in BPMN*, Hasso Plattner Institute, Potsdam, Germany, 2009, http://oryx-project.org/pub/Public/AhmedAwad/Enabling_Resource_Assignment_Constraints_in.pdf.
- [70] L. Stroppi, O. Chiotti, and P. Villarreal, *Extended Resource Perspective Support for BPMN and BPEL*, ClbSE, London, UK, 2012.



Hindawi

Submit your manuscripts at
www.hindawi.com

