Computer Science and Information Systems 22(3):783–814 https://doi.org/10.2298/CSIS240701031R

An MDA-based Requirements Analysis Process for Service-Oriented Computing Applications

Laura C. Rodriguez-Martinez[1], Hector A. Duran-Limon[2], Francisco Alvarez-Rodriguez[3], and Ricardo Mendoza-González[1]

[1] Tecnológico Nacional de México/IT Aguascalientes, Av. Adolfo López Mateos 1801 Ote., Bona Gens, CP 20256, Aguascalientes, Ags., Mexico
laura.rm@aguascalientes.tecnm.mx, mendozagric@aguascalientes.tecnm.mx
[2]CUCEA, University of Guadalajara, Periférico Norte 799, Zapopan, Jalisco, Mexico
hduran@cucea.udg.mx
[3] Autonomous University of Aguascalientes, Av. Universidad 940, Ciudad Universitaria, CP 20131, Aguascalientes, Ags., Mexico
fjalvar@correo.uaa.mx

Abstract. We propose an MDA-based requirements analysis process for Service-Oriented Computing Applications (SOCA). Our process is based on an analysis that identifies the most relevant elements of previous proposed requirements-processes. From the reviewed requirements-processes we identify such elements in terms of phases, activities, products, and roles/viewpoints. We reviewed proposals that include or emphasise the *process definition*, the *definition of products and models*, and *service-oriented modeling issues*. Also, we selected proposals within different research areas, namely Software Engineering (SE), Model-Driven Architecture (MDA), and Service-Oriented Computing (SOC). We carried out such analysis of previous requirements-processes by employing a comparative framework. We also studied some surveys about new proposals that define processes in MDA-based approaches. The main contribution of this work is a general *requirements analysis process for SOCA* called SOCA-rap that includes its activities and products allocated and grouped over a general development process. This general development process is structured in two dimensions where the first dimension involves four general activities, namely Requirements, Design, Construction, and Operation. The second dimension includes the three MDA models, namely the Computational Independent Model, the Platform Independent Model, and the Platform Specific Model. Additional contributions of this paper include (i) the identification of the phases, activities, products and roles/viewpoints of the processes of previous approaches of requirements analysis, (ii) a comparative framework of such elements, and (iii) the identification of the products included in the MDA models of the general development process.

Keywords: Service-Oriented Computing (SOC), Requirements Analysis, Architectural Design, Requirements Engineering, Model-Driven Architecture (MDA), Rational Unified Process (RUP), Service-Oriented Computing Applications (SOCA), Service-Oriented Software Engineering (SOSE).

1. Introduction

In the Software Engineering (SE) research area there are many traditional process-oriented software development models (or methodologies). As the term "process-oriented" indicates, each model describes the way to execute the development process by at least defining their phases / activities and the order to execute them. Also, such traditional process models are better structured when they define most of their elements (i.e. phases, activities, products, and roles [19]). On the other side, a special non-process-oriented development model, specifically Model-Driven Architecture (MDA) is a framework that proposes carrying out software development as a set of model refinements. The transformations between models become first class elements of the development process; therefore, a great deal of work takes places in these transformations [5]. In the process-oriented approaches the transformations between models "are implicitly defined by skilled architects in a software project" [5]. Thus, the MDA framework eases both the modelling and the model-to-model transformation. Although the MDA framework has been employed in industry and academia for software development, no specific process development model has been adopted. Also, MDA does not provide a "best practice guidance on how to maintain synchronized models on a large-scale development effort" [5]. Currently, when a practitioner uses MDA, he/she must define -in an explicit or implicit way-, a process for working (i.e., the practitioner must define a process-oriented MDA approach, commonly for a domain specific use) [5].

A process-oriented approach and an MDA framework complement each other into a ***process-oriented MDA approach*** -i.e., a hybrid approach-, by describing or capturing the model transformations explicitly integrated with a process definition. Specifically, the MDA approach for software development facilitates the analysis and transformation of the system models from the problem domain model to the executable model. Then, if we address the MDA models following an order, i.e., first the Computation Independent Model (CIM), next the Platform Independent Model (PIM) and finally the Platform Specific Model (PSM), the first problems to solve are related to the CIM.

Such an integration improves the way to develop software by merging the explicit way for modelling and performing model transformation of the MDA approaches with the explicit way for guiding the process of traditional software development process (SDP) models in SE. However, some works such as [20] recommend such an integration in both senses by either aggregating models -MDA issues- in a life cycle of software development or defining processes -SE issues- in an MDA framework for developing software. This is challenging because the integration requires great expertise in processes and modelling. As explained in [5] "*The spirit and intent of MDA is an approach that encourages use of models as the basis for software development. The ultimate vision of MDA is that models are used through the life cycle, with formal transformations between model refinements. In practice, that vision can only be realized by a very small percentage of software development organizations: the visionaries.*"

Also, Service-Oriented Software Engineering (SOSE), a parallel research area of SE, regards the study of the service-oriented paradigm [25]. In SOSE, several process models for developing Service-Oriented Computing Applications (SOCA) have been defined. Some of them focus on business modelling, system design or the use of new emergent technology [22], and a few of them cover either all the SOC applications-development life cycle or the process details [25, 26]. Specifically, the CIM modelling of a SOCA presents two big problems: 1) what to model? i.e., what are the products that represent a CIM for SOCA; and 2) how to model it?, that is, the process to develop a SOCA. On the other side, there exists a big problem to solve when we use MDA to develop software, specifically, the issue that face practitioners for defining an integration of a process-oriented approach with an MDA approach. We believe that a SOSE methodology can be defined as a process-oriented MDA approach. Hence, the general problem of defining a SOSE methodology implies several specific problems like the definition of the detailed processes of each development phase -requirements, design, construction, and operation- [22] for SOCA. Also, the general problem of defining an MDA-based SOCA development approach has specific problems such as defining the products of each MDA model and determining when such models must be developed throughout the development process.

This paper reviews scientific literature that, in the last years [20, 3], reports some surveys and proposals that study how to integrate the definition of the processes in new MDA approaches. This paper also studies the elements of Requirements-processes covering approaches of MDA, SOSE, and SE reported during the years from 1976 to 2019. And finally, we propose an MDA-based Requirements Analysis Process for SOCA that emphasizes the activity of modelling as well as defining the process that is needed to construct the products, and a way to iterate over the process activities. Model-Driven Software Development (MDSD) is a more generic approach than MDA, which involves a process. This process includes the following steps [31]: requirements modelling, model transformation (PIM to PSM), code generation (PSM to code), testing, deployment, and maintenance and evolution. However, the steps conceived for this process are rather general and do not provide detailed activities of each of the steps or the input and output artifacts of each activity. Thus, we adopt the transformation phases of MDA and propose a detailed Requirements Analysis Process for SOCA. Furthermore, the way to iterate over the proposed process activities is through the use of a generic software development process. This development process is structured in two dimensions. The first dimension involves four general activities, namely Requirements, Design, Construction, and Operation. The second dimension includes the three MDA models, namely the Computational Independent Model, the Platform Independent Model, and the Platform Specific Model.

The remainder of this paper is structured as follows. Section 2 defines a conceptual framework for the analysis of elements (i.e. phases, activities, products, and views) of the process requirements models. In Section 3, the framework of the previous section is applied to identify the elements of the ten selected requirements-processes -specially their products-; resulting in the classification and generalization of the products of the requirements processes under study. Section 4 proposes the products of the requirements processes grouped into the MDA models (CIM, PIM, PSM). Section 5 proposes a new MDA-based Requirements Analysis Process for SOCA. Section 6 presents an example of using the new MDA-based Requirements Analysis Process for SOCA. Section 7 presents an evaluation of the proposed process for SOCA. Finally, some concluding remarks are given in Section 8.

2. Framework for the Requirements-Analysis Process

Our framework for the ***Requirements-Analysis Process*** includes the following. First, it defines one criterion that includes a way for the *Classification of requirements process models.* Next, our framework describes how we can classify the *Elements of the software-system development process* in a template (see Table 1); this for the analysis of the elements of each specific requirement process under study. Finally, the framework proposes *three big viewpoints for classifying the products of the requirements processes,* and a template (see Table 2) based on such viewpoints to compare the products of the proposals.

2.1 Classification of requirements process models

The selected proposals focus on CIM modelling and offer requirements and architectural-design processes. Each proposal is classified as **RA**, **RE**, **AD**, **SOSE** and **MDA**, depending on its research area focus, respectively: *Requirements Analysis*[1], *Requirements Engineering*, *Architecture Design*, *Service-Oriented Software System*, *and Model-Driven Architecture*. Then the criterion to classify the requirements models is explained as follows.

The **RA (***Requirements Analysis***)** term refers to the methodological proposals for requirements analysis prior to the appearance of the requirements engineering research area. The term has to do with the process of analysing requirements to (i) detect and resolve conflicts between requirements; (ii) discover the links within the software and how they interact with its organizational and operational environment; and (iii) define systems requirements to specify software requirements. This has the traditional view of requirements analysis (or requirements process), which boils down to conceptual modelling using an analysis method, such as the structured analysis method. The **RE (***Requirements Engineering***)** term is widely used in the field to denote the systematic handling of requirements. This term represents a more complete vision of the requirements process that involves requirements management, an engineer of requirements as a role, and the classification and negotiation of the requirements. The **AD (***Architectural Design***)** term refers to the methodological proposals where software design activities are involved in the requirements process. Such proposals implicate that it is impossible the total separation of the two tasks -i.e., requirements and design processes are not disjoint-. In most cases the software engineer acts as a software architect too, since the processes of analysis and elaboration of requirements demands the identification of components of architecture/design that implements the requirements specification [4]. The **SOSE** (Service-Oriented Software Engineering) term refers to the methodological proposals where a requirements process is included as part of the Service-Oriented Software Systems development process.

1 Requirements Analysis -that was rather an activity-, exists before Requirements Engineering. RE emerged later on as a process that includes RA as an activity and other activities to complete the process. Afterwards, either RE substituted RA or RA was absorbed and included in RE as an activity.

The **MDA** (Model-Driven Architecture) term refers to the methodological proposals that considers or studies the requirements process as part of the Model-Driven Development approaches.

2.2 Elements of the software-system development process

For the analysis of the components of the proposals, we consider three components of the software-system development processes (SSDP) [19]. The first two components are the ***phases*** and the ***activities*** of the SSDP, which properly define the process. The third component regards the ***products*** that are constructed and interchanged during the SSDP through the phases and activities. Also, four components, defined by [4, 6, 12, 13], that are widely interrelated are considered in the analysis. These components are *view, viewpoint, stakeholder, and abstraction level*. Where, a ***view*** is made up of all the models that have to do with a specific viewpoint (point of view), represented at a certain level of abstraction or limited to a user perspective (or stakeholder). A ***viewpoint*** is a technical abstraction that focuses on a particular set of concerns that are part of the system, removing irrelevant details -and also, a viewpoint can be represented by one or more models-. A ***stakeholder*** is a person who has to do with the System in some way, as a user, the owner, the administrator, etc. The ***abstraction level*** refers to the level of detail in which a System model is represented.

Furthermore, a view can consist of one or more products. Also, each product can be represented by a specific point of view or stakeholder at different levels of abstraction. They are important components since they need to model the requirements model, in a way that facilitates its transformation into a design model and maintain the traceability between both models [2]. Table 1 presents a template for the analysis of the requirements processes under study. This template includes some numbered characteristics or aspects that are (1) the type of proposal, that is classified in RA, RE, AD, MDA, SOSE; (2) the name of the proposal; (3) the bibliographical reference of the proposal; (4) the components of the proposal (phases, activities, products) regarding the requirements process that the proposal defines in an implicit or explicit way; and (5) viewpoints, stakeholders, views, and abstraction levels that the proposal considers for representing and/or dividing the requirements.

Table 1. Framework of component-analysis of processes

Type of proposal: (1)			
Name of the proposal: (2)			
Reference: (3)			
Phase of the software-development process (4)	Products (4)	Activities (4)	Viewpoint /Stakeholder / view / abstraction level (5)
	-	-	

2.3 Three big viewpoints for classifying the products of the requirements processes

To classify, compare, and normalise the products of the requirements processes under study, we use the framework presented in Table 2. This framework is based on three big viewpoints that we consider cover the typical concerns of the products of the requirements processes. These big viewpoints are the *business viewpoint*, the *system viewpoint* and the *project viewpoint*. Where the ***business viewpoint*** involves elicitation & analysis and covers the viewpoint of the user. The ***system viewpoint*** regards modelling and covers the viewpoint of the system analyst. And the ***project viewpoint*** regards project planning and risk management issues. Then for the normalisation of the products of the requirements processes, the products are classified into these three viewpoints.

Table 2. Comparative framework to compare and normalise the products of requirements-processes

Viewpoint	P1	…	Pn
Business (elicitation & analysis, user viewpoint)			
System (modelling, system analyst viewpoint)			
Project (project planning, risk management)			

We present the complete framework, although the scope of this study does not include products and activities of the project viewpoint since they are not aligned with any MDA model.

3. Analysis of Ten Requirements-Processes

First, we present some relevant information of each proposal, highlighting the products proposed and presenting examples of how the identified products can be classified into the three viewpoints: business, system and project. Table 3 presents the list of the requirements proposals (or requirements-process models) that we analysed, referred to as P1..P10. First, we present some relevant information of each proposal, highlighting the products proposed and presenting examples of how the identified products can be classified into the three viewpoints: business (elicitation & analysis, viewpoint of the user), system (modelling, viewpoint of the system analyst) and project (project planning and risk management).

The proposals reviewed covers the software development processes for modelling a CIM. These proposals are commonly either requirements or architectural design processes. Then, through the analysis of such proposals, we identify and classify their activities, products, and the views for the representation of the product models.

Table 3. Proposals of CIM modeling and of requirements processes

Proposal	Reference	Type of proposal	Name of the proposal
P1	Leite, 1988 [12]	RA	"Viewpoint Resolution in Requirements Elicitation"
P2	Boehm, 2004 [27]	RA	(MBASE)
P3	Getchell, 2002 [29]	RE	"MSF Process Model v. 3.1"
P4	Hofmeister, 2005 [8]	AD	"Global Analysis"
P5	Hofmeister, 2007 [9]	AD	"General Model of Software Architecture Design Process"
P6	Péraire, 2007 [21]	SOSE & RE	(RUP-SOA)
P7	Habe, 2013 [7]	AD	"The Missing Link -between Requirements and Design"
P8	Bourque, 2014 [4]	RE	(general model proposed in SWEBOK V3.0)
P9	Asadi, 2008 [3]	MDA	"An MDA-based System Development Lifecycle"
P10	Rodríguez-Martínez, 2019 [22]	MDA & SOSE	(SOCA-DSEM)

Below we present a brief description and analysis of the elements of each reviewed proposal.

P1. "Viewpoint Resolution in Requirements Elicitation" [12]. In [12] Leite (1988) proposes a process-oriented definition of RA: "*Requirements analysis is a process which '**what is to be done**' is elicited and modeled. This process has to consider different viewpoints, and it uses a combination of methods, tools and actors. The product of this process is a model, from which a document, the requirements is produced*". This work proposes two documents as products: a "Requirements analysis model" and "Requirements document". The first document is defined from the viewpoint of the analyst, is internal to the development process, is written in a machine-processable form, and must be readable by analysts and designers to understand the application to build. The second document reflects the viewpoint of the user, is in agreement with the user of what he/she demands from the system and must be readable by both users and analysts [12]. Such documents correspond to two viewpoints, specifically the "business" and "system" viewpoints.

P2. "MBASE" [27]. It is an approach that integrates the process-models, products, properties, and the success criteria to develop a software system. The essence of the approach is to develop some elements of systems definition in a concurrent and iterative way (or by refinement) by using the Win-Win Spiral model of Boehm [27]. Such elements are the following: Operational Concept Description (OCD), System and Software Requirements Definition (SSRD), System and Software Architecture Description (SSAD), Life Cycle Plan (LCP), Feasibility Rationale Description (FRD), Construction Transition Support (CTS) plans and reports, and Risk-driven prototypes. Issues of risks are implicit in MBASE since it is a risk-driven approach. The risk control is implicit in the way to proceed, i.e., it does not indicate products or documents of risk planning or management, e.g., it uses risk-driven prototypes.

P3. "MSF Process Model V.3.1" [29]. In [29], Getchell proposes several products in two phases: (i) the Envisioning Phase, is where the project is defined and planned; and (ii) the Planning Phase, is where requirements are specified and the software development is planned. In such phases several products are proposed, for the three big viewpoints. For example, for the single activity in the Envisioning Phase it has the following products for each viewpoint: (1) for the business viewpoint: proposes the "vision/scope document", (2) for the project viewpoint: proposes the product "risk assessment document and project structure document", and (3) for the system viewpoint: proposes the product of "prototypes". Also, in the case of the Planning Phase the proposal has the product "Business, user, operational and systems Requirements List" in the business viewpoint.

P4. "Global Analysis" [8]. Hofmeister in [8] proposes the activity "Global Analysis" as an intermediary between the processes of the requirements analysis and the architecture design. Such a Global Analysis helps to guide the design process by capturing the design rationale and supporting traceability between requirements and architecture. This proposal has three products. The product Software Requirements Specification (SRS) belongs to the business viewpoint. The other two products, namely the Architecture Design Requirements (ADR) and the System Solutions Strategies (SSS) are part of the system viewpoint. Although ADR and SSS are constructed in the Global Analysis, ADR is part of the requirements specification whereas SSS is part of the architecture specification. ADR is located on the problem (requirements) side. Such an architecture must identify those requirements that affect the architecture design and also, must identify any additional quality attributes and constraints (e.g., pre-imposed design decisions and limitations, use of COTS software). It is also important to identify other factors that affect the architecture (e.g., culture of the organization, state of technology, existence of a related product line). On the other side, SSS is a set of strategies that guide the architecture design.

P5. "General Model of Software Architecture Design Process" [9]. This proposal defines the activity of Architectural Analysis (AA) that is an intermediary activity between the requirements analysis and the architectural design. Products of AA are: (i) Context Requirements and (ii) Architectural Significant Requirements. The requirements analysis presents the product "Context Requirements" for the business viewpoint and the product "Architectural Concerns" for the system viewpoint.

P6. "RUP-SOA" [21]. RUP-SOA [21] is an extended version of RUP for SOA (o RUPz). RUPz has two dimensions. Each dimension corresponds with an axis. The first dimension is the ***time*** represented over the horizontal axis. The second dimension corresponds with the ***disciplines*** or activities represented over the vertical axis. The horizontal axis represents also, the software development life cycle that is divided into four phases: Inception, Elaboration, Construction, and Transition. Each phase is divided in one or more iterations, this, according to the project needs. In addition, the vertical axis represents disciplines, such as requirements, analysis, design, or implementation, logically grouped according to its nature. The requirements products of RUPz are classified into the viewpoints of "business", "system" or "project": e.g., for the business viewpoint there is the product "Business Case", for the system viewpoint there is the product "Use-Case Model", and for the project viewpoint there is the product "Test Strategy".

P7. "The Missing Link -between Requirements and Design" [7]. In this proposal, Habe [7] describes Product Abstraction Levels (PAL) aligned with the V-Model. This shows the transformation Trail of the information and shows how the product is developed by keeping consistency and completeness, despite the high-fragmented activities of the engineering and the increasing parallelization of the sub-processes (like requirements-process) of the Product Creation Process (PCP). This proposal also tells us how the requirements are interconnected and that the requirements process is not integrated with the design process. The requirements product involves transforming the "stakeholder requirements" to the "functional and product requirements" which in turn are transformed to the "system requirements". The latter are then transformed to the "technical requirements of implementation" which are then transformed to the "implementation".

P8. "General model defined by the SWEBOK V3.0" [4]. Bourque (2014) [4] proposes a product for three big viewpoints: e.g., the product "System Definition Document" for the business viewpoint, the product "System Requirements Document" for the business and system viewpoints, and the product "Software Requirements Specification Document" for the system viewpoint and "prototype" for the system and business viewpoints.

P9. "MDA-Based System Development Lifecycle" [3]. Asadi [3] proposes a software-system development life cycle (of the software development process) based on MDA. The proposal comprises five phases: Project Initiation, PIM Development, PSM & Code Development, Deployment, and Maintenance. Of such phases, the phase PIM Development and the phase PSM & Code Development are typically executed in an iterative and incremental way. This approach proposes products for two viewpoints: business and project. But the requirements process (project initiation) does not include products for the big system viewpoint, i.e., the approach advocates a separate vision to construct requirements and design.

P10. "SOCA-DSEM" [22]. The proposal P10 is an MDA approach that is focused on the development of SOSS and indicates what products are part of CIM, PIM, PSM, and PM. Although SOCA-DSEM proposes CIM products [29], it does not consider activities for requirements engineering. The proposed activities are more like sub-phases. It proposes that each MDA model should be constructed thoroughly in a phase or activity. It also proposes products, and artifacts to be constructed in a specific notation. The requirements phase has products that can be classified as the business and project viewpoints.

Regarding the component-analysis of the proposals of requirements-processes, Section 4 presents the normalisation of the products identified for each of one of the proposals under study.

4. Products of Requirements process classified into CIM, PIM or PSM

This section proposes the products that are part of the models CIM, PIM or PSM. We propose a comparative framework -that is shown in Table 2- to compare the products of the proposals. The framework enables the normalisation of the products. This normalisation involves identifying the products that can be part of a CIM, PIM or PSM.

We use the conceptual framework to identify the components, specifically regarding the products that compose the ten proposals under study. Then, given the identified products, we carried out a normalisation (or generalization) of the products on the requirements-process. Table 2 presents a comparative framework, in which the first column indicates the three viewpoints (business, system, and project) and the next n columns present the products of each proposal. The result of such a normalisation is presented in Table 4 and Table 5 involving the products of the business and system viewpoints, respectively. The general products of the requirement process, identified in the previous section, are indicated in Table 4 and Table 5 with a their names and a label i.e., CIM-1..CIM-6, PIM-1..PIM-2, PSM-1.

Table 4. Normalisation of products oft he business viewpoint

Products maped from Requirements proposals reviewed regarding Business viewpoint										**SOCA-rap proposal CIM model**	
P1 RA	**P2 RA**	**P3 RE**	**P4 AD**	**P5 AD**	**P6 SOSE & RE**	**P7 RE**	**P8 RE**	**P9 MDA**	**P10 MDA-SOSE**	**products**	**docu-ments**
- Facts - Application Vocabulary	- Stakeholders list - Win-Win conditions	Vision/scope document	Software requirements specification (SRS)	X	Business case	X	System Requirements Document	Computational Independent Model (CIM)	X	CIM-1. Vision/scope (or businesss case document) CIM-2. Stakeholders list CIM-3. Win-win conditions CIM-4. Context (Facts & Application Vocabulary)	**System Definition Document**
Requirements document	X	Requirements List		Conceptual Business Analysis	X	- Operational Use level requirements - Functional level requirements			Business Process Model (BPM)	CIM-5. Requirements List CIM-6. Business Process Model (BPM)	**System Requirements Document**

As shown in Table 4 and Table 5:

(i) The products of the requirements processes are classified into the business viewpoint and are aligned with CIM. Also, the products of requirements processes, classified into the system viewpoint are aligned with PIM and PSM.

(ii) The products of the Requirements-process that were identified as part of the CIM, PIM, PSM models are: for CIM are CIM-1, .. , CIM-6, for PIM are PIM-1 and PIM-2 and for PSM is PSM-1.

The first version of the Requirements documents implies evolution during the development process. It is assumed, that the CIM documents and the document “Software Requirements Specification” enable requirements changes. Requirements evolve along all the development phases if there are changes on the requirements specification. In contrast, the "Initial Design Document" is assumed to be disposable in the early stages of the development process as it is about quick experiments with some non-detailed technological alternatives.

Table 5. Normalisation of products of the system viewpoint

Products maped from Requirements proposals reviewed regarding System viewpoint										**SOCA-rap proposal PIM and PSM models**	
P1 RA	**P2 RA**	**P3 RE**	**P4 AD**	**P5 AD**	**P6 SOSE & RE**	**P7 RE**	**P8 RE**	**P9 MDA**	**P10 MDA-SOSE**	**products**	**docu-ments**
Requirements analysis Model	- Operational Concept Description (OCD) - System and Software Requirements Definition (SSRD)	Functional Specification	X	X	- Vision - Use-Case Model - Supplementary Specifications - Glossary	Systems Requirements	Software Requirements Specification Document	Requirements Model	Enterprise architecture	PIM-1. Functional Specification of software (includes a minimal enterprise architecture for services)	**Software Requirements Specification (SRS)**
	System and Software Architecture Description (SSAD)	Design document (initial)	Architecture Design Requirements (ADR)	- System Context of architectural concerns - Architectural Concerns - Architecturally significant requirements (ASR)	Software Architecture Document	- Functional architecture - Systems architecture	X	X	System Solution Architecture: (- Service Provider design, - Service Consumer design, - Database design)	PIM-2. System Architecture Solution (includes service provider design, service consumer design, database design)	
	Prototype	Prototypes, development and technological options, feasibility analysis	X	X	X	Models and design options	Prototype	X	Preliminary user interface design as a prototype	PSM-1. Prototypes, development and technological options, feasibility analysis and preliminary user interface design	**Initial Design Document**

Finally, Table 6 describes the identified products of the CIM, PIM and PSM as follows.

Table 6. CIM, PIM and PSM products of requirements-process for SOSS

CIM products	
CIM-1. Vision/scope (or business case document).	A high-level vision of the objectives and constraints of the project.
CIM-2. Stakeholders list	Indicates names commonly as roles of the stakeholders, and the description of the function and tasks of such roles.
CIM-3. Win-Win conditions	It Specifies the stakeholder win-win relationship throughout the system's life cycle. It involves conditions related to the stakeholders to reach success by getting the stakeholders to clarify, understand, and reconcile their success models.
CIM-4. Context (Facts & Application Vocabulary)	This regards the context of the system but mainly the Facts and the Application Vocabulary (or domain terms). A Fact refers to gathering information for better understanding an application. More specifically, it refers to the domain knowledge that can be gathered from documents, pre-printed forms, information between stakeholders, interviews with stakeholders, etc. The Application Vocabulary standardises the use of a common set of words in the model of the application and in the application itself.
CIM-5. Requirements List	This involves the Operation Use level requirements and Functional level requirements that are agreed with the user [7]: Operational Use level requirements captures the customer needs, the operational use terms and conditions, product roadmap, product scope, and the development-process conditions. Functional level requirements capture the properties, functions and constraints of the system and how to validate and ensure the behaviour and feasibility.
CIM-6. Business Process Model (BPM)	It includes the definition of the process and workflows at the business level and the identification of the services and their conceptual definition.
PIM products	
PIM-1. Functional Specification (includes a minimal Enterprise Architecture for Services)	This regards a set of requirements specification in machine-processable form that includes an early design [13] (i.e. the Functional Specification and the Enterprise architecture). The Functional Specification describes in detail how each feature (or requirement) looks like and behaves. It also describes the architecture and the design of all features [29]. It has Instructions for the developers on what to build, the Basis for estimating work, the Agreement with customer on exactly what will be built, and the Point of synchronization for the whole team. The Enterprise architecture is a logical structure for classifying and organising the descriptive representations of an Enterprise that are significant to the management of the Enterprise as well as to the development of the Enterprise's systems, manual systems, and automated systems [24].
PIM-2. System Architecture Solution (an initial architecture design of service provider, service consumer and database design)	It is the candidate architectural solution that may present alternative solutions, and/or may be partial solutions (i.e., fragments of an architecture). It reflects design decisions about the software structure. The architectural solutions include information about the design rationale, that is, commentary on why decisions are made, which decisions are accepted or rejected, and the traceability of decisions to requirements is defined. It refers to the initial or candidate solution, constructed in the first development phases [22] that includes the initial Service operations definition for the service provider design, and the semantic specification of the processes and workflows for the service consumer design; and the initial database design such as an Entity and Relation diagram.
PSM products	
PSM-1. Prototypes, development and technology options, feasibility analysis.	Prototypes, development and technology options, feasibility analysis [29] refers to selecting and proving the technology and an initial design of the system through prototypes for analysing the feasibility of constructing the system with the selected technology. And the initial interface design.

5. The SOCA-rap Process

SOCA-rap is a general MDA-based process that integrates MDA models for Requirements modelling and a process to guide the development. Such a process is represented in the Figure 1 and includes the four generic phases of a software development life-cycle (SDLC) i.e. Requirements, Design, Construction, and Operation. The process also includes the models of MDA -CIM, PIM, and PSM- and an extension the model E-PM (Executable-Platform Model) [14]. Similar to the RUP process, the area below the curves indicates the amount of development effort required in each phase to construct each model. The number of X on the general MDA-based process (see Figure 1) indicates the average amount of effort required for constructing each MDA model in each phase. The SOCA-rap then defines the MDA products (see Table 6), the model-to-model transformations (see Figure 3) and the generic activities of the requirements analysis process (see Table 7) to be specifically used in the SOCA domain. The SOCA-rap also defines a very generic SDLC for iterative and incremental development (see Figure 1).

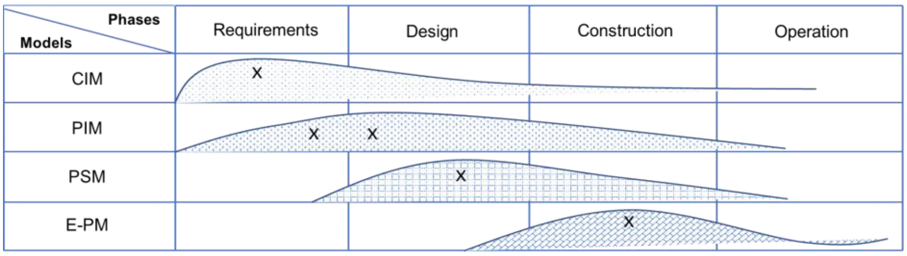

Fig. 1. General MDA approach of an SDLC

The distribution of activities and products is shown in Figure 2, which also presents the summary of our proposed MDA-based Requirements Analysis Process.

In Table 7 we define the general activities of the Requirements process presented in Figure 2, namely “Elicitation”, “Analysis”, “Modelling”, and “Validation”. We associate the products identified in the previous section (see Table 6) with such general activities. The three activities “Elicitation”, “Analysis”, and “Validation” corresponds to three of the four Knowledge Areas - “Requirements Elicitation”, “Requirements Analysis”, “Requirements Specification”, “Requirements Validation”-, defined in [4]. Here we assume that the requirements specification is executed along all the requirement process. The products of the activities of Elicitation and Analysis are presented in terms of the business viewpoint. The names of such products have a prefix that indicates the MDA model to which each one belongs. Hence, the following products constitute the CIM model: CIM-1 Vision/scope (for the business case document), CIM-2 Stakeholders list, CIM-3 Win-Win conditions, CIM-4 Context (Facts & Application Vocabulary), CIM-5 Requirements List and CIM-6 Business Process Model (BPM). Besides, the PIM model includes the product PIM-1 Functional Specification (which includes a minimal Enterprise Architecture for Services) and PIM-2 System Architecture Solution. Finally, the PSM model for requirements includes only the product PSM-1 Prototypes, development and technology options, and feasibility analysis.

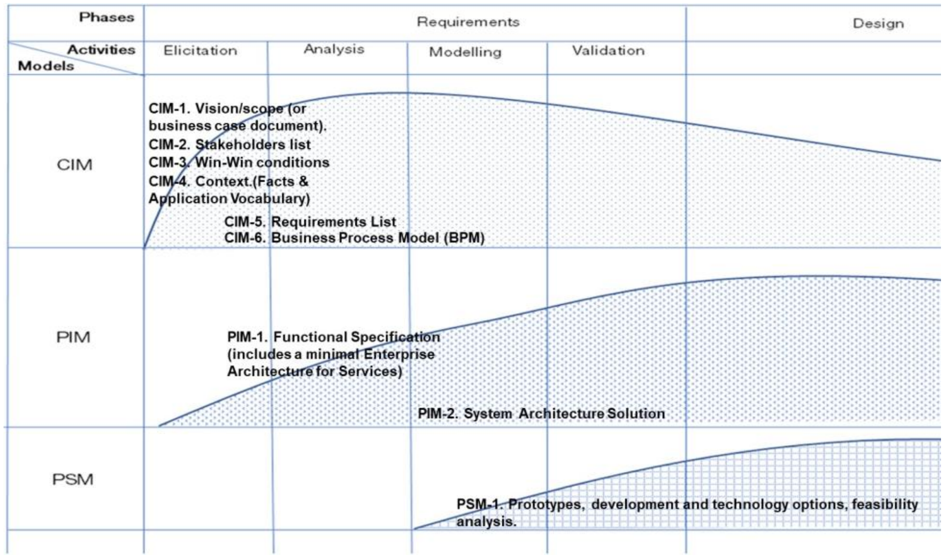

Fig. 2. MDA-based Requirements Analysis Process

The process is executed as follows. First, the requirements are taken during the activity of elicitation and analysis from the problem domain and then modelled (or transformed) as a CIM that is presented from a business viewpoint. Next, the modelling activity transforms the CIM into a PIM from the system viewpoint. Such a system viewpoint involves a model at a high-level of abstraction. At this time, we have the definition of requirements from the user and system perspectives [12]. Afterwards, the activity of modelling transforms the PIMs into PSMs. A PSM represents the requirements from the perspective of computing and involves a pre-design, e.g., it is expected that the PSM model includes some design of interfaces. Finally, the CIM, PIM, and PSM products are validated in the “Validation” activity.

We obtained the description of the requirements-process activities mainly from [4] and by matching with the description of the reviewed proposals -specially the descriptions of Elicitation, Analysis, and Validation were obtained from the methodology P10, and of the description of Modelling was obtained from the methodology P1-. Table 7 shows the description of the activities of the requirements process.

Finally, as shown in Fig. 3, the methodology uses three lines of model-to-model transformations [22], namely orchestration, choreography, and data lines. The first products constructed are CIM-1, CIM-2, CIM-3, and CIM-4, which are transformed into the CIM-5 Requirements List and the CIM-6 Business Process Diagram –involving both the Choreography and Orchestration lines-.

Also, in the Data transformation line, it is constructed a new version of CIM-4 Context by aggregating the Enterprise Data view or Application Vocabulary. Then the CIM products are transformed into the PIM products as follows. The products CIM-5 and CIM-6 in the Choreography line are transformed into the PIM-1 Functional specification (Use-case details). Also, CIM-5 and CIM-6 in the Orchestration line are transformed into the first version of the PIM-2 System Architecture Solution that includes a System model diagram. Then PIM-1 in the Choreography line is transformed in a second version of PIM-2 that includes the Join Realization Table. Next, the second version of CIM-4 is transformed into a third version of PIM-2 that includes the E-R System data model. Finally, in the Choreography transformation line the third version of PIM-2 is transformed into PSM-1.

Table 7. Activities of the Requirements-Process

Activity	Description
Requirements Elicitation	It is "…concerned with the origins of software requirements and how the software engineer can collect them. It is the first stage in building and understanding of the problem the software is required to solve. It is fundamentally a human activity and is where the stakeholders are identified, and relationships established between the development team and the customer" [4].
Requirements Analysis	It is "… the process of analyzing requirements to detect and resolve conflicts between requirements; discover the bounds of the software and how it must interact with its organizational and operational environment; elaborate system requirements to derive software requirements. The traditional view of requirements analysis has been that it be reduced to conceptual modeling using one of a number of analysis methods, such as the structured analysis method" [4].
Modelling / Specification	Although the activities of Modelling and Specification are realised in parallel with each activity of requirements process, it is indicated here a modelling activity to highlight the modelling activity given that the specification part remains only as a parallel activity. Here the Modelling activity is considered a transition phase between the domain specification -conceptual model or domain model- and the systems requirements. In this activity, the system begins to be modelled/designed and the CIM-5 Requirements List, CIM-6 Business Process Model (BPM), PIM-1 Functional Specification and PIM-2 System Architecture Solution are completed. Also, in this activity is where it is initiated the elaboration of the products PSM-1 Prototypes, development and technology options, and feasibility analysis. Also, according to Leite (1988) [12] "*The resultant requirements analysis model, faces the problem that it has to serve different actors. First, it has to be readable by users and second, it should be the base for the designer's understanding of the application.*" Finally, the authors in [4] state that "The development of models of a real-world problem is key to software requirements analysis. Their purpose is to aid in understanding the situation in which the problem occurs, as well as depicting a solution. Hence, conceptual models comprise models to reflect their real-world relationships and dependencies. Several kinds of models can be developed. These include use diagrams, data flow models, state models, goal-based models, data models, and many others."
Requirements Validation	"The requirements documents may be subject to validation and verification procedures. The requirements may be validated to ensure that the software engineer has understood the requirements; it is also important to verify that a requirements document conforms to company standards and that it is understandable, consistent, and complete" [4].

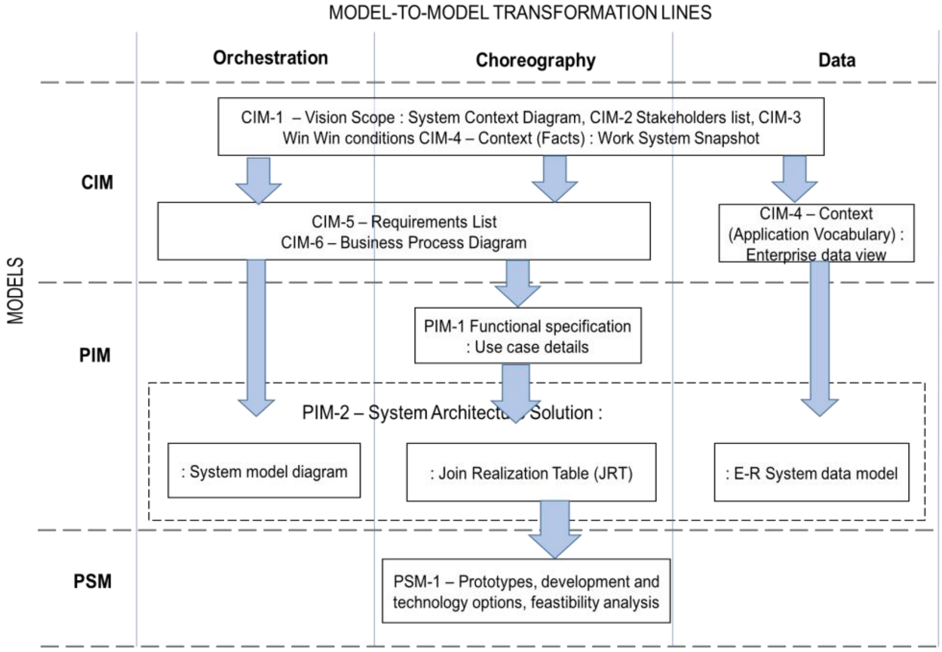

Fig. 3. Model-to-model transformation of SOCA-rap

SOCA-rap states how the model-to-model transformation process should be carried out. However, the details of how to carry out these transformations are out of the scope of this paper.

6. Illustrative Example

In this section, we illustrate the proposed requirements process by applying it to an example case. The illustrative example involves a system for managing job competences (JCM System- Job Competences Management System). The objective of this JCM System is to enable the users for managing such job competences, e.g. definition of competences, record of job competences of the personnel (or employees) of the organization, managing job descriptions by competencies, evaluation of job competences, and define career plans and training plans for the employees. The products of the CIM, PIM and PSM developed for SOSS, with the requirements-analysis process, are presented in the illustrative example. The notations used for the example, are optional, i.e. they are used only for illustration. Then, any other alternative notations can be used.

6.1 Computational Independent Model for the example

The product CIM-1 Vision Scope presents a high-level vision of the objectives and constraints of the project (see Figure 4). The notations for CIM-1 are from SOCA DSEM[2] [22] and UML[3] [34].

The product CIM-2 Stakeholders list indicates the names commonly used as roles of the stakeholders, and the description of the functions and tasks of each role; but, it is not included in this example for the sake of brevity. The product CIM-3 Win-Win Conditions is optional, hence, here it is not exemplified. In this case, we suggest using the approach of Boehm & Kitapci (2006) [28]. The product CIM-4 Context (Facts & Application Vocabulary) is exemplified in the Figure 5, which presents a work system snapshot[4] [32] and an excerpt of the application vocabulary as in [22].

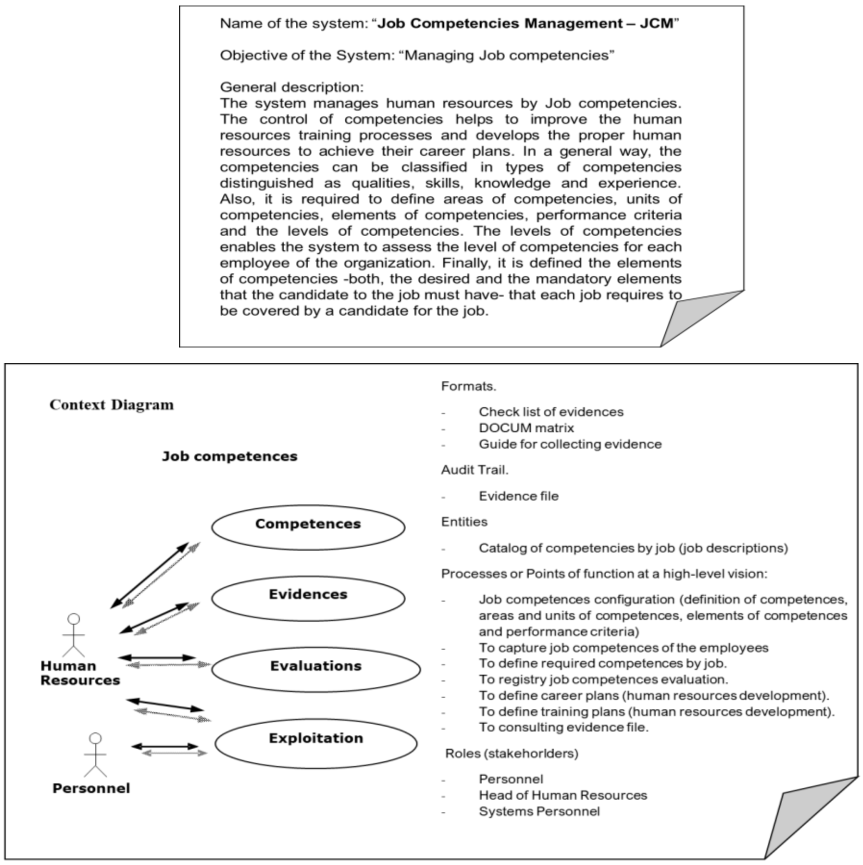

Fig. 4. CIM-1 Vision Scope

----
2 SOCA DSEM is a Software Engineering Methodology to develop Service-Oriented Computing Applications

3 UML is a Unified Modelling Language widely accepted by software engineering academicians and practitioners

Customer	Products and services
Personnel Head of Human Resources	Audit trail consulting and validation To configure job competencies, areas and units of competence, competency elements and levels of competence. To capture performance criteria for evaluating competences Scheduling training plans To define career plans To capture grades of competence evaluations To capture evidences of audit trail To consult career and training plans

Processes and activities
1. When the employee is hired, the Human Resources Head captures grades of competencies for a job of the employee and the respective evidences as an audit trail. 2. At the end of the first year of hiring, the Human Resources Head generates a Training and Career Plan of the employee . 3. From the second year, once in a year, the Human Resources Head captures for the employee new competencies, and the corresponding grades and evidences. The employee can generate changes on a training plan. 4. Each two years, the employee or the Human Resource Head can change the career or training plan. 5. At every moment, the Human Resources Head can consult candidates for a vacancy based on the grades of competencies of the employees for reaching career planes, independently of the current career plan.

Participants	Information	Technologies
Personnel Head of Human Resources Systems Personnel	- Audit trail information (grades, evidence, and real dates of evaluation) - Systems personnel database - Career Plans - Job descriptions - DOCUM matrix - Catalog of competencies - Evaluation criteria - Calendar of training	Database management Internet Worksheet for data exportation HTML templates

Definitions, acroyms and abbreviations

Standard of competence
The norm or standard of competence is an orderly description of the content of a work performance or work achievement. The standard states:
• Elements of competence.
• Performance criteria.
• Evidence of performance.
• Scope (field of application).
• Guide for collecting evidence of performance.

Elements of competence
It is the description of work achievements, grouped as units of competence. Each element of competence must be described in terms of: a verb (the action), an object (on which the action is performed), and the condition, which indicates the result of the action on the object. This includes required knowledge.

Performance criteria
It is a quality criterion with which the result of evaluation of the element of competence must be achieved. That is, it involves sources of verification on the quality of work achievement that the competence allows to obtain. They are classified as performance evidence (see types of performance evidence).

Evidence of performance.
It is the evidence that the achievement has been effectively achieved, this is under the established performance criteria.

Scope (field of application).
Machines, tools, physical context, environment and interpersonal re-operation in which the scope performance is verified.

. . .

Fig. 5. CIM-4 Context (Facts & Application Vocabulary) represented as Work System Snapshot and Enterprise data view

4 Work System Snapshot is a format proposed by Alter [32] to show the relationship between business processes, their resulting products and services, their IT support and the stakeholders, from a business perspective.

Next, the product CIM-5 Requirements List presents, in a contextual way, the functions of the system to be constructed (see Figure 6). Figure 6 presents the objectives of the system and the requirements organised in system development stages. In addition, Figure 7 presents the contextual diagram of use cases of the system. Parts a and b of CIM-5 are presented in a non-specific or pre-established format. Part c is presented in the UML use case notation.

Project Name: "Job Competencies Management – JCM"
Date: October 3th, 2005
Period: August-December, 2005

Objectives.

The computarised system of Job competencies aims to support the standardisation, training and certification of Job competencies within the Organisation that offers employment contracts (Ooec).

Contributing to raise the qualification of the human resources of the Ooec and consequently to generate better conditions of competitiveness of the personnel.

It will help in the effective planning of the training according to the requirements of the Ooec.

At the level of individuals, the implementation of work skills favors the mobility of workers within the Ooec and allows them to prove their knowledge to obtain the recognition of their performance, regardless of whether they have obtained them through training or in The job.

The scope of Job competencies implies a substantial improvement in human resources management, reducing their personnel selection and training costs and facilitating their link with professional training systems.

The general objectives priority is as follows:
1. Dynamically configure the catalog of job skills (definition of competences).
2. Register the job skills of employees.
3. Define the job skills required by each position as part of their job description.
4. Evaluate the job skills of employees.
5. Define career plans based on competencies.
6. Generate training plans.

Project Name: "Job Competencies Management – JCM"
Date: October 3th, 2005
Period: August-December, 2005

Requirements by stages of development of the system

The first Stage includes the functions:
- Definition and printing of competencies catalog.
- Record of competency assessments.
- Definition of the catalog of positions with their descriptions based on competencies.
- Personal skills report.
- Report of competent personnel by position.

Note: only the functions of this first stage are described in this version of the requirements document.

The second stage includes the functions:
- Recording, consulting and printing of evidence.
- Assessment report.
- Report of competent personnel by position.
- Planning of competence assessments.

The third stage of development includes the functions:
- Recording, consulting and printing of evidence of performance.
- Consulting and printing of training plans.
- Consulting and printing of evaluation results.
- Definition, consulting and printing of career plans.

In the fourth stage of development, an extension will be made that will allow grouping evidence of performance to assist in the evaluation registry.

The maintenance stage will also include the series of exploitation reports that are required, reaching a good collection of statistical reports.

Fig. 6. CIM-5 Requirements list, part a: objectives and part b: requirements in system development stages

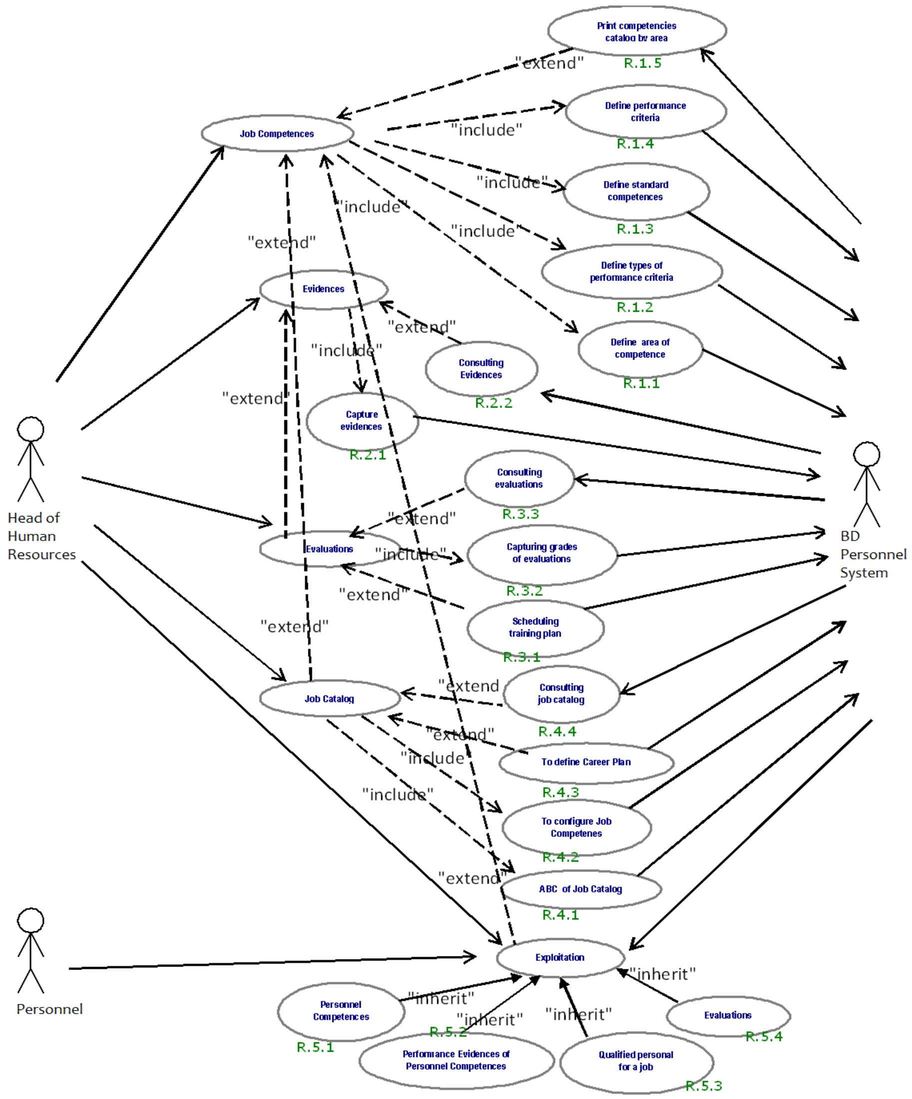

Fig. 7. CIM-5 Requirements list, part c: contextual diagram of use cases of the system

Finally, the last product of the model is CIM-6 Business Process Model. This model is presented in different perspectives - part a, b, and c (see Figures 8, 9 and 10)-. Figure 8 presents the private processes (part a), Figure 9 shows the process collaboration (part b), and Figure 10 depicts a detailed process diagram for the actor "Head of Human Resources". CIM-6 is presented in BPMN[5] as in [22].

5 BPMN is a Notation for Business Process Modelling

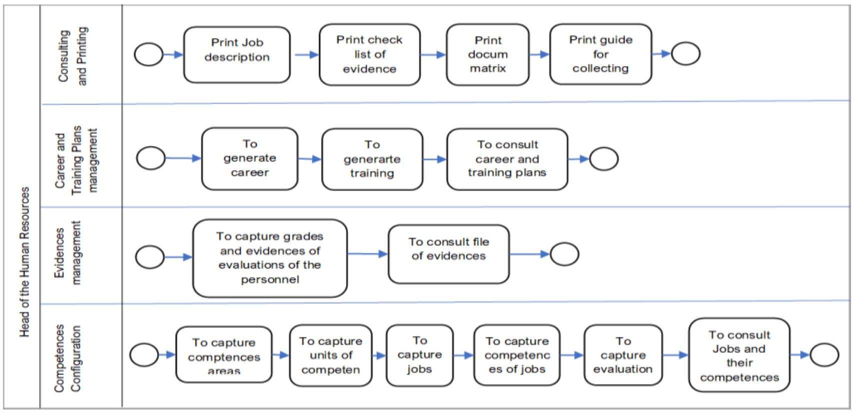

Fig. 8. CIM-6 Business Process Model (BPM), part a: Business process diagrams – private processes

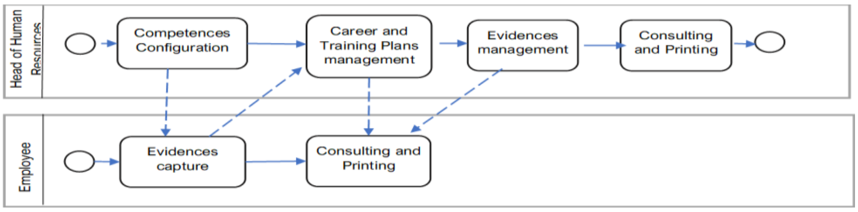

Fig. 9. CIM-6 Business Process Model (BPM), part b: process collaboration (actors' collaboration)

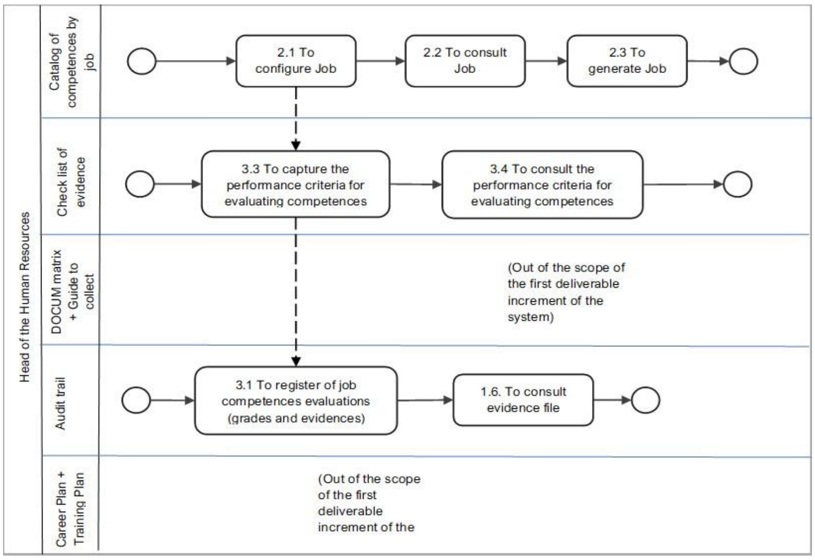

Fig. 10. CIM-6 Business Process Model (BPM) part c: A detailed business process diagram (workflows of the actor "Head of Human Resources")

6.2 Platform Independent Model for the example

For the product PIM-1 Functional Specification, one example, for a use case (see Table 8). It should be noted that each use case requires a Table as part of the product PIM-1. For PIM-1, this example uses the use case description format used in an example of RUP[6] in [33].

Table 8. Use case R.1.1 Define the competence area

Use case	R.1.1 Define competence area
Actor	Head of Human Resources
Purpose	Register and keep the competence areas updated in the competence catalogue.
Summary	The use case is initiated by the user who captures the data that identifies and describes the area of competence. The system records the data captured by the user. The user can modify or delete the data of an area of competence while maintaining the consistency and integrity of the system.
Exceptions	The system will indicate the user that due to the integrity and consistency of the data; an area of competence cannot be eliminated. In the same way, the system will validate the values of the area of competence according to the data rules defined below.

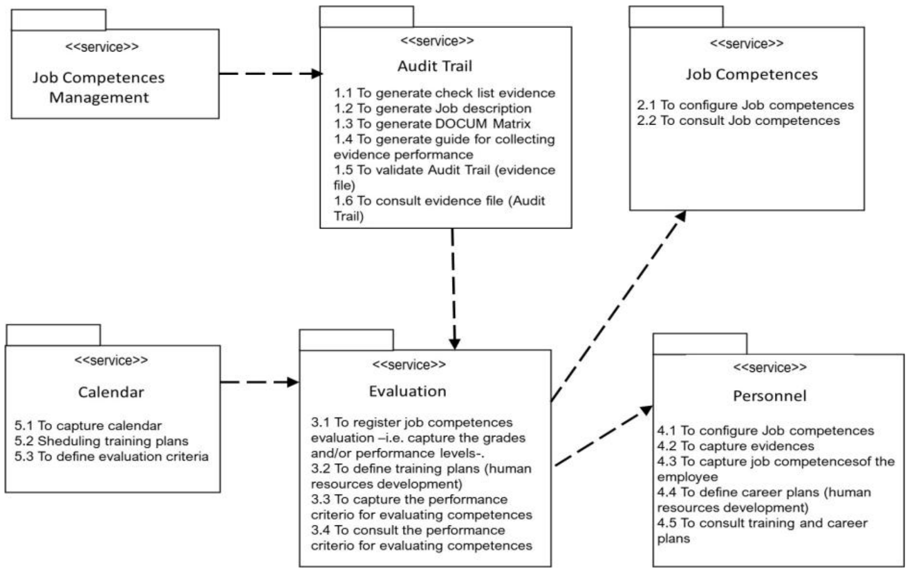

Fig. 11. PIM-2 System-Solution Architecture part a: Service Model diagram (regarding provider and composition of services)

---
6 RUP – Rational Unified Process

Table 9. PIM-2 System-Solution Architecture part b: JRT – Joint Realisation Table of use cases (regarding the choreograghy) for use case R.1.1 Define competence area

The use case R.1.1 Define competence area		
Action of the Actor	**White Box**	**Service and operation to use**
This use case starts when the Head of Human Resources selects the options of Competences / Define competence area	The system will display a text box and a search button. If the user knows the key competence area it is looking for: they can capture in the text box the code of the corresponding competence. Otherwise, you can search for a specific competence area by pressing the search button.	Job Competences service, operation 2.2 To consult Job competences.
When capturing the area key	The system will automatically search for the required data and display it. If the system does not find the password, the system will assume that it is desired to register a new password, at this point the user may cancel the registration operation or continue with the registration of the new area.	Job Competences service, operation 2.1 To configure Job competences.
The user presses the search button	The system will show a dialog that enables to do a contextual searching by the key competence, description or any other significative field for such a competence area. After locating a specific area of competence, the system will display the data of the area in question. At this point the user can indicate the system if he/she wants to modify the data of the area or if he/she wants to delete them.	Job Competences service, operation 2.2 To consult Job competences.

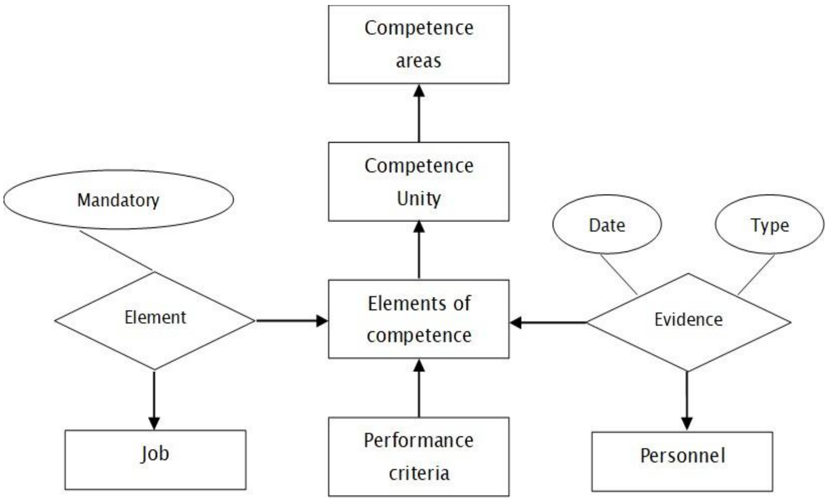

Fig. 12. PIM-2 System-Solution Architecture part c: Entity-Relationship diagram for the first stage or increment of development of the system

The product PIM-2 System-Solution Architecture includes an initial architectural design of the service provider and the service consumer (initial definition of composition, orchestration and choreography), and an initial design of the database (see the Figure 11 and the Table 9). The initial version of the architecture design has the need of refine, correct, and complete the design in order to achieve a better version on subsequent development phases. More specifically, Figure 11 presents the Service Model diagram (regarding service provider and choreography), Table 9 present the Join Realization Tables, each one for one of three use cases (regarding consumer and orchestration), and Figure 12 presents the Entity-Relation diagram (regarding the database). Regarding the notations used for the PIM-2 example, part a is presented as a stereotyped diagram for services as in [22]. Part b is a Joint Realisation Table taken from RUP in [33] and applied as in [22]. Part c is an Entity Relationship Diagram [35].

6.3 Platform Specific Model for the example

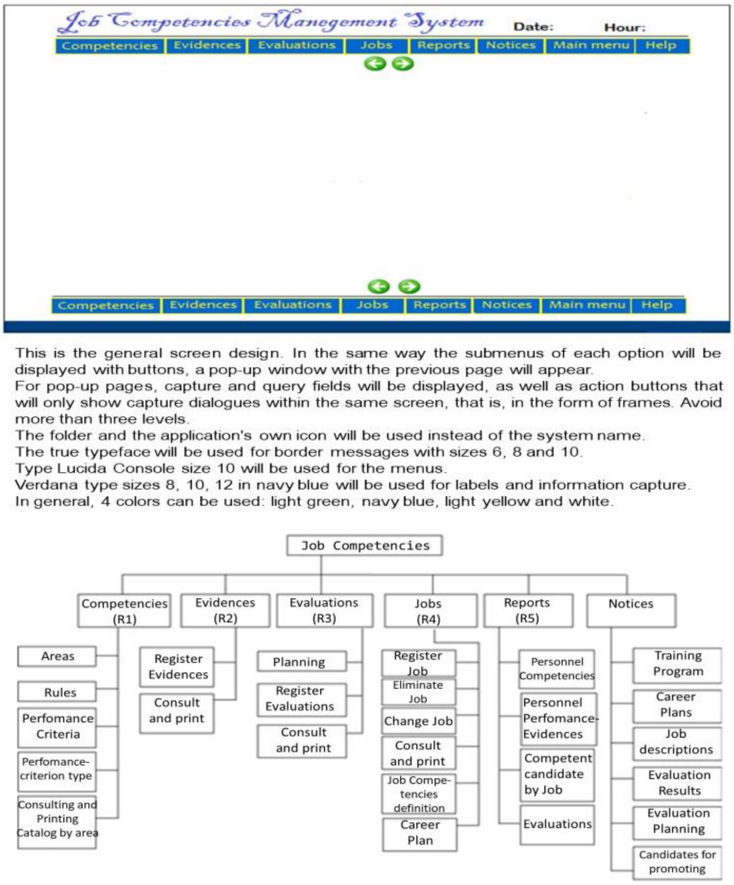

Fig. 13. PSM-1 Prototyping, part a: interface design of the main screen and part b: navigational design of the system

Finally, Figures 13 and 14 present the product PSM-1 Prototyping that involves the construction of prototypes for designing interfaces and prove the possibility of connection for the initial architecture design of provider and consumer. This PSM-1 also proposes technology options [29] for analysing the feasibility of constructing the system in any of them. The illustrative example presents only the preliminary design of the main screen of the system and a proposal of navigation between the options of the system. More specifically, Figure 13 presents the initial interface design of the main screen of the system. Here we use paper prototypes as in [22] to illustrate the PSM-1 Prototyping.

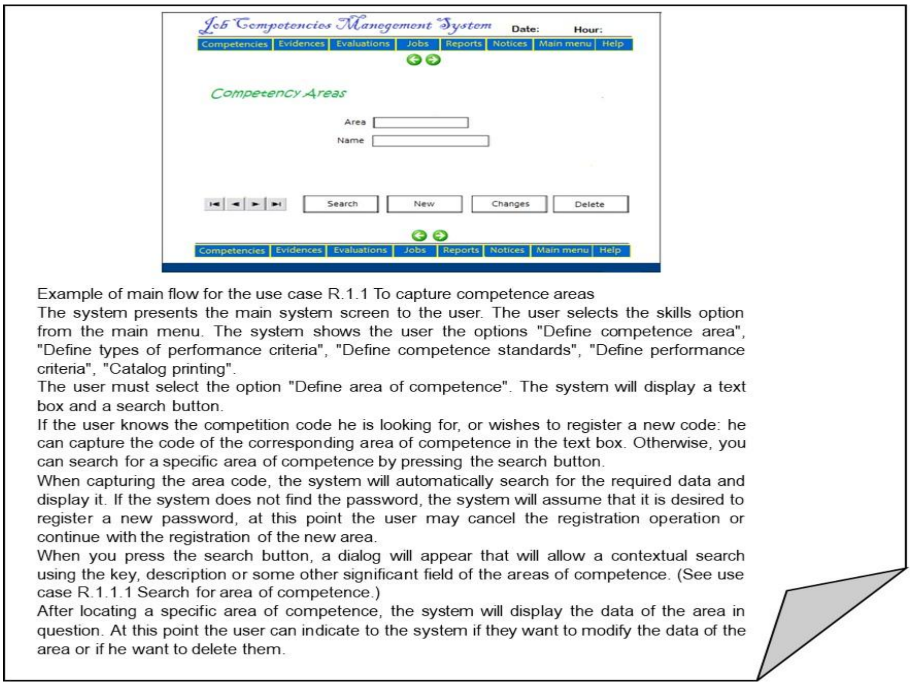

Fig. 14. PSM-1 Prototyping, part c: interface design for register, eliminate or change a competence area as a standard interface for the system

7. Evaluation of the proposal

The reviewed proposals present an evolution on the knowledge of the requirements. The first type of proposals, the RA proposals, regard the activity of “requirements analysis”. In contrast, in the RE proposals the requirements are managed as a process called “requirements engineering”. Then, the AD proposals add new characteristics whereby the relationship between “requirements analysis” and “system design” is described and eases the transformation of the requirements analysis into the system analysis. Consequently, the SOSE proposals adapt the requirements activities to construct

Service-Oriented Computing Applications. Finally, the MDA proposals change the process, i.e., it proposes a different way to construct a system focusing on the models to be constructed during the requirements process.

We claim that a solid requirements analysis processes for SOCA should provide support for RE, MDA, and SOSE. The reason of this is described as follows. In the early days, RA was incorporated and diluted into RE. Nowadays, AD is also being incorporated into RE. In this context, AD, in an analogous form, highlights the importance of the architecture design closely linked to the requirements specification. In addition, the research areas, namely RA, AD and MDA, highlight their respective link between (1) the requirements process and the system design, (2) the requirements and the architectural design; and (3) the products categorised in models with their approach to designing and transforming. In turn, RE provides the model outlining the activities to be performed during the requirements process, some products to be built during each activity, and the process management activities. However, RE, does not describe how to construct or design the products. MDA addresses this by defining a way to construct various evolving models throughout the development process, encompassing the requirements process. Finally, with the current trend toward web-based systems, constructed via services or microservices, it is required to focus on the construction of SOSE products.

We evaluate our approach by comparing it to the reviewed proposals. Table 10 shows the type of support provided by each proposal. The third, fourth, and fifth columns of Table 10 indicate whether each proposal supports RE, MDA, and SOSE, respectively, through a check mark for positive support or a cross for no support.

Table 10. Comparative analysis of the support provided by the proposals

Proposal & its name	Type of proposal	RE support	MDA support	SOSE support
P1 – Leite, 1988 [12] -"Viewpoint Resolution in Requirements Elicitation"	RA	√	×	×
P2 – Boehm, 2004 [27] - (MBASE)	RA	√	×	×
P3 – Getchell, 2002 [29] - "MSF Process Model v. 3.1"	RE	√	×	×
P4 – Hofmeister, 2005 [8] - "Global Analysis"	AD	√	×	×
P5 – Hofmeister, 2007 [9] - "General Model of Software Architecture Design Process"	AD	√	×	×
P6 – Péraire, 2007 [21] - (RUP-SOA)	SOSE & RE	√	×	√
P7 – Habe, 2013 [7] - "The Missing Link - between Requirements and Design"	AD	√	×	×
P8 – Bourque, 2014 [4] - (general model proposed in SWEBOK V3.0)	RE	√	×	×
P9 – Asadi, 2008 [3] - "An MDA-based System Development Lifecycle"	MDA	√	√	×
P10 – Rodríguez-Martínez, 2019 [22] - (SOCA-DSEM)	MDA & SOSE	×	√	√
SOCA-rap - integrated MDA-based requirements process - MDA & SOSE RE process	MDA - SOSE - RE	√	√	√

We analyse below the elements that are supported by the proposals for each type of support i.e. RE, MDA and SOSE.

7.1 Support for RE

Currently, RE refers to proposals that encompass both RA and AD. Initially, RA proposals focused primarily on the business point of view. Subsequently, RE proposals shifted focus towards process-oriented approaches. Currently, RE proposals tend to emphasise the system point of view, specifically regarding architectural design. Furthermore, ***requirements modelling*** and ***early system design*** remain relevant in the literature. P1, P5, and P6 approach modelling and early design from the ***software architecture research area*** [12], [8], [7], which has significantly expanded in the last decade. As a result, innovative proposals are needed to deal with the intersection of requirements and architecture. In essence, RA deals with the modelling of business requirements, while RE evolves describing the process derived from RA. The emerging AD then transforms business requirements into system design with modelling playing a vital role. And, RE emphasises the process but also considers modelling from both RA and AD.

All proposals, except P10, provide support for RE since they support the requirements analysis in either forms: (1) as requirements-process definition, or (2) as requirements-modeling strategy (e.g. CIM modeling with MDA). The first form mainly defines the process whereas the second one mainly establishes the products. Given that P1 and P2 are presented as definitions of a requirement-processes, they naturally support RE. Also, P3 to P8 inherently support RE as they are derived from RE and its corresponding AD. Regarding R9, it offers the two forms, as a result, it supports RE too. Lastly, P10 is presented as requirements-modeling strategy, therefore P10 does not support RE. Regarding *SOCA-rap*, it provides support for RE by defining the activities and their description, and by establishing how these activities iterate throughout the process. More specifically, SOCA-rap includes the activities of *Requirements Elicitation*, *Requirements Analysis*, *Modelling/Specification*, and *Requirements Validation* (see Table 7). The way to iterate in SOCA-rap consists of two dimensions related to the whole SDLC (see Figure 1). The SDLC is divided into four phases, namely Requirements, Design, Construction and Operation, which make up the horizontal dimension of the process. Each phase comprises specific activities. For instance, the SOCA-rap activities pertain to the Requirements phase. Meanwhile, the vertical dimension deals with products and entails building MDA models during specific phases. Table 6 shows each model that encompasses various SOCA-rap products.

7.2 Support for MDA

Support for MDA refers to proposals that focus on Model-Driven Development. Such proposals focus on defining products to be built that are grouped in the CIM, PIM, PSM and Executable Models, as well as the transformation between these models. Nevertheless, current approaches also strive to detail the development process for each stage of the software development process. The CIM model is closely interconnected with the requirements stage. Also, the PIM, PSM and Executable model are closely related to the architectural design stage, the detailed design stage and the implementation stage, respectively.

Only P9, P10, and SOCA-rap provide support for MDA. P9 provides an SDLC description for developing MDA models. It is advised to construct each model in a singular phase, with corresponding activities and products allocated to each stage, respectively. Regarding RE, P9 propose to construct two products, namely "*CIM model*" and the "*Requirements model*," but does not provide details. P10 proposes stages, activities and products for a comprehensive SDLC process for developing service-based systems. The proposal does not encompass the requirements analysis process. Instead, it puts forward the CIM model for service-based systems. On the other hand, SOCA-rap proposes the requirements analysis process by outlining the activities and their products. These activities are grouped into the Requirements phase. Additionally, SOCA-rap categorises the products into the CIM, PIM and PSM. As the requirements process has an obvious connection to the CIM model, the majority of the products fall into this category. Although, there is a typical association of requirements with the CIM model, some products of the Requirements phase are not Computing Independent. As a consequence, these products should be included into the PIM or PSM model. SOCA-rap considers such grouping emphasizing the early design and the linkage of requirements and architecture design.

7.3 Support for SOSE

SOSE support pertains to processes that take into account service-oriented development. P10 proposes the products of the CIM model that are directly related to RE. Therefore, only P6, P10, and SOCA-rap provide support for SOSE. P6 introduces a RUP version for service-based systems, which describes each phase of the software development process. P10, on the other hand, focuses on the products of requirements and modelling questions but does not address the requirements phase process. In addition, SOCA-rap includes the necessary elements, and specially, the products mandated by SOSE requirements. Furthermore, P6, P10, and SOCA-rap address requirements, either through the definition of the requirements process definition and/or definition of requirement products.

SOCA-rap includes and describes most of the products that are present in RE and in SOSE proposals (see Table 6). The products are presented as part of the different MDA models; specifically, those that have products related to the requirements process and those models that are constructed during the requirements process. Such models are CIM, PIM, and PSM.

Overall, we can see that the proposals that provide better support are P6, P10, and SOCA-rap. However, the two former miss to provide support to MDA and RE, respectively, whereas SOCA-rap provides full support to all elements of Table 10.

8. Threats to Validity

In the case of threats to internal validity, the results of the accuracy evaluation could be affected by a bias in the selection of the ten works that were used to compare our work. We reduced this threat by following disciplined principles to search the reviewed works. These principles are: 1) the work explicitly or implicitly define artifacts to be developed during the requirements process, 2) the work define the activities of the requirements process, design process and/or CIM modeling, 3) the work defines its products in terms of viewpoints for the various stakeholders in the business or development process. These search principles privilege the manual analysis of the works. The manual analysis allowed us to identify and discriminate themes or topics, and refine searches. Furthermore, it allowed us to consider references whose documents are difficult to find even with the complete reference, especially, old references of doctoral theses or technical reports.

Threats to external validity are related to the extent the fulfillment of the evaluated aspects reflect the most important characteristics of a RA process for SOCA. We improved external validity by using an example case, which showed the feasibility of our approach. The example case shows how the requirements process builds the CIM, and also shows how to build the first stages of the PIM and the PSM. The artifacts constructed exemplify the requirements products of each MDA model according with the viewpoints proposed by SOCA-rap. The example case, also illustrates how, following the transformation lines in Figure 3, it is possible to transform some products into others. And how these transformations, progressively link the domain model (CIM) to the system models (PIM and PSM). Where the PIM is still independent of how it will be implemented, and the PSM is already a design to be implemented.

It is also shown that the definition of the artifacts is applicable to SOCA. The case example illustrates how to follow the SOCA-rap process, its transformation lines, and the construction of its products. This illustrates how each model in the transformation line is aligned first to business issues, then to system issues, and then to the implementation. Finally, the example case illustrates the execution of the first iteration of the development process for the requirements. Finally, the construction of the product CIM-5 provides a rough idea of the number of iterations/increments (i.e. stages of development) that are required to complete the system.

9. Conclusion

We methodologically studied ten approaches, which come from related complementary areas, namely RA, RE, AD, SOSE, and MDA. Our study found some characteristics that a good requirements analysis process for SOCA should possess. These characteristics are the following: (1) the process has an early design on the software development, (2) it considers the link between requirements analysis and architectural design, and (3) it describes the products and the models to be constructed. We then employed and applied a comparative framework to the reviewed methodologies in order to identify the elements that a SOCA requirements analysis process needs to meet the above-mentioned characteristics. Based on this, we developed SOCA-rap, which defines its elements in terms of phases, activities, products, and roles/viewpoints. SOCA-rap covers all characteristics, except the means to carry out model transformation among the models. Hence, SOCA-rap defines the products of the requirements process. Each product is defined as part of an MDA model. In addition, each model is documented in specific documents of requirements. Concretely, the CIM is documented in the System Definition Document and the System Requirements Document, while the PIM and PSM are documented in the Software Requirements Specification (SRS) and the Initial Design Document.

In section 7 we evaluated the support that SOCA-rap and the reviewed methodologies provide to RE, SOSE, and MDA. We can see that SOCA-rap offers more support than the other approaches.

Finally, as future work, we will provide a model-to-model guide for carrying out model transformation of the models defined in the SOCA-rap.

References

1. Aguilar, J.A., Garrigós, I., Mazón, J.: Requirements Engineering in the Development Process of Web Systems: A Systematic Literature Review, Acta Polytechnica Hungarica, 13(3). (2016)
2. Amna, N., Anam, A., Farooque, A.: Model Driven Architecture – Issues, Challenges and Future Directions, Journal of Software, 11(9), 924-933. (2016)
3. Asadi, M., Ravakhah, M., Ramsin, R.: An MDA-based System Development Lifecycle, Proceedings of Second Asia International Conference on Modeling & Simulation, IEEE Computer Society, 836-842. (2008)
4. Bourque, P., Fairlley, R.E.: SWEBOK V3.0 Guide to the Software Engineering Body of Knowledge, IEEE Computer Society, 1-335. (2014)
5. Brown, A.W.: Model driven architecture: Principles and practice, Springer Verlag, Expert's voice. Software and Systems Modeling, 3(1) 314-327, Digital Object Identifier (DOI) 10.1007/s10270-004-0061-2. (2004)
6. Cantor, M.: Rational Unified Process for Systems Engineering Part 1, 2, 3, IBM Rational Software. (2003)
7. Habe, A., Michielsen, C.: The Missing Link -between Requirements and Design, Proceedings of the Posters Workshop at CSD&M. (2013)
8. Hofmeister, C., Nord, R.L., Soni, D.: Global Analysis: moving from software requirements specification to structural views of the software architecture, Iee Proc.-Softw., 152(4), 187-197. IEE Proceedings online no. 20045052, doi: 10.1049/ip-sen:20045052. (2005)
9. Hofmeister, C., Kruchten, P., Nord, R.L., Obbink, H., Ran, A., America, P.: A general model of software architecture design derived from five industrial approaches. Elsevier, The Journal of Systems and Software 80, 106-126. (2007)
10. Khan, S., Dulloo, A.B., Verma, M.: Systematic Review of Requirements Elicitation Techniques, International Journal of Information and Computation Technology, 4(2), 133-138. (2014)
11. Leite, J.: A Survey on Requirements Analysis, Technical Report, University of California, Department of Information and Computer Science. (1987)
12. Leite, J.: Viewpoint Resolution in Requirements Elicitation. PHD thesis, Department of Computer Science, University of California, Irvine. (1988)
13. Leite, J.: Viewpoint analysis: a case study, ACM SIGSOFT Software Engineering Notes, 14(3), 111-119. (1989)
14. Miller, J., Mukerji, J.: MDA Guide Version 1.0.1. OMG. (2003)
15. Neil, C.J., Laplante, P.A.: Requirements Engineering: The State of the Practice, IEEE Software, IEEE Computer Society, 20(1), 40-45. (2003)

18. Nuseibeh, B., Easterbrook, S.: Requirements Engineering: A Roadmap, ICSE. (2000)
19. Oktaba, H., Ibargüengoitia, G.: Software Process Modeled with Objects: Static View, Computación y Sistemas 1(4), 228-238. CIC-IPN ISSN 1405-5546. (1998)
20. Parviainen, P., Takalo, J., Teppola, S., Tihinen, M.: Model-Driven Development – Processes and practices, VTT Working papers 114. (2009)
21. Péraire, C., Edwards, M., Fernandes, A., Mancin, E., Carrol, K.: The IBM Rational Unified Process for System z, IBM Rational software, Redbooks. (2007)
22. Rodriguez-Martinez, L.C., Duran-Limon, H.A., Mora, M., Alvarez-Rodriguez, F.: SOCA-DSEM: a Well-Structured SOCA Development Systems Engineering Methodology, Computer Science and Information Systems (COMSIS), 16(1), 19-44. (2019)
23. Singh, Y., Sood, M.: Model Driven Architecture: A Perspective, IEEE International Advance Computing Conference (IACC 2009), 1644-1652. (2009)
24. Zachman, J.A.: The framework for Enterprise Architecture: Backgroud, decription and utility, https://www.zachman.com (2016)
25. Gu, Q., Lago, P.: A stakeholder-driven service life cycle model for SOA, In Proceedings of 2nd international workshop on Service oriented software engineering: in conjunction with the 6th ESEC/FSE joint meeting (IW-SOSWE'07) 1-7. Dubrovnik, Croatia: ACM. (2007)
26. Gu, Q., Lago, P.: Guiding the selection of service-oriented software engineering methodologies, SpringerLink, Service Oriented Computing and Applications (SOCA), 5(4), 203-223. (2011)
27. Boehm, B., Klappholz, D., Colbert, E., Puri, P., Jain, A., Bhuta, J., Kitapci, H.: Guidelines for Model-Based (System) Architecting and Software Engineering (MBASE), Center for Software Engineering, University of Southern California. (2004)
28. Boehm, B., Kitapci, H.: The WinWin Approach: Using a Requirements Negotiation Tool for Rationale Capture and Use. In: Dutoit A.H., McCall R., Mistrík I., Paech B. (eds) Rationale Management in Software Engineering. Springer, Berlin, Heidelberg, 173-190. (2006)
29. Getchell, S., Hargrave, L., Haynes, P., Lubrecht, M., Pervez, K., et al.; MSF Process Model v. 3.1. Microsoft Corporation, Microsoft Solutions Framework, White Paper. (2002)
30. Richards, M., Neal, F.: Fundamentals of Software Architecture: An Engineering Approach. Ed. O'Reilly. (2020)
31. Stahl, T., Völter, M., Bettin, J., Haase, A., Helsen, S.: Model-Driven Software Development. John Wiley & Sons. (2006)
32. Alter, S.: The Work System Method: Connecting People, Processes, and IT for Business Results, Work System Press. (2006)
33. Rational Software Corporation.: Rational Unified Process for Systems Engineering SE1.1, Rational Software Corporation, White paper. (2002)
34. Larman, C.: UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition, Addison Wesley Professional. (2004)
35. Silberschatz, A., Korth, H.F., Sudarshan, S.: Database System Concepts, Seventh Edition., Mc Graw-Hill. (2019)
36. Pressman, R.S.: Software Engineering: A Practitioner's Approach, Fifth Edition, McGraw-Hill series in computer science. (2001)
37. OMG.: Business Process Model and Notation (BPMN), Version 2.0.2, OMG. (2013)

Laura C. Rodriguez-Martinez is a full Professor at the Systems and Computing Department, Institute of Technology of Aguascalientes, Mexico. She holds a PhD in Computer Science at Universidad Autonomous University of Aguascalientes, Mexico in 2009. Her research interests include Software Systems Development Processes, Service-Oriented Software Engineering and Graphical-User Interfaces Development Processes.

Hector A. Duran-Limon Eng.D., has published more than 25 research papers in peer-reviewed journals listed in JCRs and more than 40 papers in international top conferences and research books. He obtained an IBM Faculty award in 2008. He holds an M.Sc. in Computer Science (1994) from the National Autonomous University of Mexico (UNAM), and a PhD in Computer Science (2001) from Lancaster University, England. Prof. Duran-Limon was a research assistant at Lancaster University (2002-2003), a Professor at the Tec Monterrey (2004-2005), and has been a full-time Professor, since 2006, at the University of Guadalajara, Mexico. Prof. Duran-Limon has directed the thesis of eight PhD students. His current research interests are software architectures, software product lines, and HPC in the cloud. Prof. Duran-Limon is also a Mexican National Researcher at Level I. Overall, Prof. Duran-Limon has more than 20 years of experience doing research, and teaching both undergraduate and graduate courses.

Francisco Alvarez-Rodríguez is a Professor of Software Engineering. He holds a BA. in Informatics (1994) and a MA. (1997) from the Autonomous University of Aguascalientes and a EdD degree from the Education Institute of Tamaulipas, México and he is PhD from the National Autonomous University of Mexico. He has published research papers in several international conferences in the topics of software engineering and e-learning process. His research interests are software engineering lifecycles for small and medium sized enterprises and software engineering process for e-learning. He is currently president of the National Council for Accreditation of programs and Computing , A.C. (CONAIC).

Ricardo Mendoza-González is a full-time professor at the Tecnológico Nacional de México/ IT Aguascalientes. Member of the National Researchers System (Level 1) from the Secretariat of Science, Humanities, Technology and Innovation, SECIHTI (in Spanish), Mexico. His current research interests include several topics on (but not limited to): Human-Computer Interaction, User Research, User-Centered Design, Usability, Accessibility and Equity in technology, Design Thinking, User Interfaces Design, Innovation processes, Educational Technology, Open Educational Resources, Software Engineering and Artificial Intelligence.

Received: July 01, 2024; Accepted: February 16, 2025.